

Università degli Studi di Bergamo  
Department of Engineering and Applied Sciences  
Ph.D. in Engineering and Applied Sciences  
XXXVII cycle



# **A framework for the real-time integration of mechatronic systems data with augmented reality applications to enhance human-machine interaction**

Advisor: Prof. Bruno Fausto Zappa  
Co-advisor: Prof. Roberto Strada

Ph.D. coordinator: Prof. Alessandra Marini

Ph.D. thesis of:  
**Jasmine Santinelli**

# Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
<b>2</b>	<b>Literature Review: Analysis and Insights</b>	<b>9</b>
2.1	Introduction . . . . .	9
2.2	Analysis of existing literature reviews . . . . .	10
2.2.1	Summary and Commentary on Existing Reviews . . . . .	10
2.2.2	Key themes . . . . .	15
2.2.3	Identified challenges and future research directions . . . . .	16
2.3	Common applications and proven implementations . . . . .	18
2.3.1	Usability and Interfacing in AR for Industrial Applications . . . . .	18
2.3.2	Machine and Process Monitoring in Industrial Applications using AR . . . . .	19
2.3.3	AR-assisted industrial maintenance . . . . .	21
2.3.4	Augmented Reality in Industrial Quality Control . . . . .	25
2.3.5	Human-Robot Collaboration and AR . . . . .	27
2.4	Conclusions . . . . .	28
<b>3</b>	<b>Framework</b>	<b>31</b>
3.1	Introduction and Motivation . . . . .	31
3.2	Architecture definition . . . . .	34
3.2.1	The Mechatronic System . . . . .	38
3.2.2	Augmented Reality Applications . . . . .	40
3.2.3	The Server . . . . .	41
3.3	Design Process . . . . .	42
3.3.1	AR Application design phase . . . . .	45
3.3.2	Mechatronic system information modelling phase . . . . .	48
3.3.3	Server design phase . . . . .	53
3.4	AR Application Development Model: Workflow Based on the Information Model . . . . .	54
3.5	Conclusions . . . . .	58
<b>4</b>	<b>Applying the framework to a case study</b>	<b>60</b>
4.1	Introduction . . . . .	60
4.2	Design Process . . . . .	62
4.2.1	AR Application design phase . . . . .	62
4.2.2	Mechatronic system information modelling phase . . . . .	67
4.2.3	Server design phase . . . . .	75
4.3	Implementation . . . . .	79
4.3.1	Server . . . . .	80
4.3.2	Mechatronic system controller upgrade . . . . .	86
4.3.3	AR Application development . . . . .	87
4.4	Conclusions . . . . .	97
<b>5</b>	<b>Results</b>	<b>99</b>
5.1	Development in Simulated Environment: Achieved Results and Foundations for a Digital Twin for Remote Monitoring . . . . .	99
5.2	Results from AR Deployment on Magic Leap 2 . . . . .	105

## List of Figures

1	Research openings for the use of AR in industry . . . . .	16
2	Information Diagram . . . . .	35
3	Information Diagram using an intermediary . . . . .	36
4	Structural diagram of the architecture, with subsystems and main components . . . . .	37
5	Flow chart of the design process . . . . .	43
6	AR Application design phase . . . . .	45
7	Mechatronic system information modeling phase . . . . .	49
8	ARObject main components . . . . .	50
9	UML class diagram of ARObject . . . . .	51
10	Comprehensive information model . . . . .	52
11	Server design phase . . . . .	54
12	System activity diagram . . . . .	55
13	AR Application detailed flow . . . . .	57
14	Cartesian robot in the Mechatronics and Mechanical Dynamics Laboratory . . . . .	60
15	Cartesian robot's 3D model . . . . .	61
16	Magic Leap 2 Headset, Compute Pack and Controller . . . . .	64
17	Magic Leap2 FOV . . . . .	65
18	Application of the information model to the case study . . . . .	68
19	3D model with Origin coordinates . . . . .	69
20	3D model with marker on some sensors . . . . .	70
21	Multi-layered architecture of OPC UA . . . . .	77
22	Architecture representation . . . . .	79
23	OPC UA information model . . . . .	82
24	OPC UA UML Specification . . . . .	83
25	UAExpert Adress space . . . . .	86
26	Magic Leap Hub 3 . . . . .	87
27	Model target generator overview . . . . .	91
28	Guide views . . . . .	92
29	Representation of perspectives for tracking the Cartesian robot . . . . .	93
30	OpenXR Structure . . . . .	94
31	Action Map HMD . . . . .	95
32	Action Map Controller . . . . .	95
33	Unity interface showing the developed application . . . . .	100
34	Scene view with all canvases enabled . . . . .	101
35	Scene view with the main Canvas . . . . .	102
36	Game view of Figure 35 with main Canvas . . . . .	102
37	Centralized Controller User Interface, parameters . . . . .	103
38	Centralized Controller User Interface, properties . . . . .	103
39	Details on the position of the AR markers . . . . .	104
40	Details on the proximity sensors . . . . .	104
41	Linear encoder and motor details . . . . .	105

42	Application result on Magic Leap2: front view . . . . .	106
43	Application result on Magic Leap2: front view with controller . . . . .	107
44	Application result on Magic Leap2:side view . . . . .	108
45	NI 6229 and motor details . . . . .	109
46	Linear encoder State Tab selection . . . . .	109
47	Starting Task3 using the controller UI . . . . .	110
48	Interacting with the AR marker: Activating the X-axis linear encoder canvas with the controller . . . . .	110
49	Using the main menu: Enabling the second acquisition board UI in the components section . . . . .	111

## List of Tables

1	Technological Summary of AR Implementations . . . . .	30
2	Functional and Non-Functional Requirements for Industrial AR Applications . . . . .	46
3	Translation of Functional Requirements into Technical Specifications . . . . .	47
4	Definition of AR Application Requirements for Mechatronic System Monitoring . . . . .	62
5	Requisite AR Application and Magic Leap 2 Specifications . . . . .	63
6	Technical Specifications of Magic Leap 2 . . . . .	66
7	X and Y axis motor data . . . . .	71
8	Linear Encoders data . . . . .	72
9	National Instruments PCI 6229 I/O Boards data . . . . .	73
10	Proximity Sensors Data (Axis Y) . . . . .	73
11	Proximity Sensors Data (Axis X) . . . . .	74
12	Tasks Data . . . . .	75
13	Comparison between OPC Classic and OPC UA . . . . .	76
14	OpenXR Action maps details . . . . .	96

# 1 Introduction

Augmented Reality (AR) is emerging as a innovative technology with vast potential across various industrial sectors. By overlaying digital information onto the physical world, AR enhances perception and interaction with the environment, opening new avenues for innovation, efficiency, and sustainability. As industries transition into the eras of Industry 4.0 and Industry 5.0, AR is set to play a pivotal role in reshaping operations and promoting sustainable practices. This makes research in the field of AR not only timely but crucial for understanding and maximizing its benefits.

Industry 4.0 aims to transform manufacturing processes by integrating advanced technologies to create smarter, interconnected, and more adaptable systems. In this context, Augmented Reality (AR) plays a crucial role in enhancing operational efficiency and flexibility. AR enables the overlay of digital information onto the physical world, providing immediate access to critical data and operational instructions. This helps companies optimize maintenance, streamline production workflows, and improve decision-making by offering real-time insights and support directly where they are needed. Through these capabilities, AR contributes to achieving the core objectives of Industry 4.0, such as increased productivity, reduced downtime, and enhanced adaptability to changing demands.

As we advance towards Industry 5.0, which focuses on seamless collaboration between humans and machines, the role of Augmented Reality (AR) becomes increasingly pivotal. Industry 5.0 emphasizes personalization, sustainability, and human-centric approaches, aiming to create environments where technology enhances human potential rather than replacing it. AR supports this vision by enabling intuitive human-machine interactions, providing workers with tools to visualize and interact with complex data more naturally, and fostering adaptive and personalized production processes. By integrating AR, industries can achieve greater customization, improve worker well-being, and adopt more sustainable practices, aligning with the core objectives of Industry 5.0.

Industry 5.0 emphasizes not only technological advancements but also environmental and social responsibility. In the context of Industry 5.0, Augmented Reality (AR) contributes to sustainability not only by optimizing industrial processes but also by improving working conditions. By making complex tasks more intuitive and accessible, AR can reduce the physical and cognitive load on workers, enhancing their overall well-being and safety. This leads to greater productivity and a healthier work environment, which is central to the concept of social sustainability. In addition, by streamlining processes and reducing the need for resources, AR promotes more sustainable practices within industries, aligning with goals such as those outlined in the PON REACT-EU program, which focuses on fostering both economic and social resilience.

Having outlined the potential of augmented reality in the modern industrial context, and given the growing interest in conducting research in this area, an extensive literature review was carried out to evaluate the current state of AR technology, along with its capabilities and limitations. The review confirmed the potential of AR in various industrial applications. It also identified three primary categories of factors that limit the widespread adoption of AR in industrial settings: technological factors, which include the limitations of current AR devices in terms of performance, accuracy, and usability; user factors, such as ergonomics, user experience, and the cognitive load associated with AR applications; and industrial factors, which encompass organizational readiness, integration challenges, and the cost-benefit analysis of adopting AR technologies.

While much research has been dedicated to addressing technological and user-related factors, industrial factors, particularly integration challenges, remain less explored. Existing studies often provide case-specific solutions that demonstrate AR’s potential in assisting specific tasks, but they fall short of offering generalized, scalable solutions for broader industrial adoption.

Key obstacles that must be overcome to support AR integration and address broader industrial challenges include the integration of AR with existing machinery, where clear methodologies for retrofitting into established systems are often lacking. Another challenge is data representation, as mechatronic systems generate large amounts of information that need to be correctly represented in AR applications, requiring a structured approach to manage and display this complexity effectively. Furthermore, the lack of a universal architecture that is independent of specific devices or applications limits the adaptability of AR solutions in various industrial environments. The absence of a standardized development procedure for AR monitoring applications adds further complexity for industries seeking to implement AR. Lastly, scalability and interoperability remain significant hurdles, as many current AR solutions are designed for specific use cases and require substantial modifications to be adapted to different settings.

This work aims to address the challenges of integrating AR into industrial environments by presenting a comprehensive and innovative framework for developing AR-based human-machine interaction systems. These systems enable the real-time monitoring of components and processes, as well as the control of the system itself. The framework provides a structured and scalable approach that not only resolves the identified obstacles but also offers clear guidelines for implementing AR applications in industrial contexts. In this context, the term "framework" refers to a systematic methodology designed to streamline the development and deployment of AR-based systems. It serves as a blueprint to guide the integration of AR into industrial workflows, ensuring consistency, scalability, and efficiency throughout the implementation process.

The framework is designed to address the key obstacles related to AR integration in industrial settings, as identified in the previous paragraph. It consists of three main components: the design procedure for defining the architecture and its components, the information representation model, and the AR application development model—workflow based on the information model. Each of these components directly contributes to solving the challenges of integrating AR into existing industrial workflows.

- **Integration with Existing Machinery:** The framework addresses this challenge by defining an architecture based on an intermediary server, which facilitates the integration of AR applications with existing systems. This approach allows for seamless integration without requiring substantial modifications to the existing applications. Instead, the relevant data is exposed in real-time to the intermediary server, which manages the communication between the AR application and the legacy systems, ensuring smooth interoperability without the need for extensive reengineering of the original systems.
- **Data Representation:** The framework’s focus on developing an object-oriented model for data representation addresses the issue of managing and displaying complex data generated by mechatronic systems. By creating a structured approach to representing data, the framework ensures

that the vast amounts of information produced by mechatronic systems are effectively organized and displayed within the AR environment. This makes it easier to manage data in a way that is both meaningful and easy to interpret for industrial users.

- **Lack of Universal Architecture:** The framework provides a solution to the lack of a universal architecture by defining a generic, adaptable structure for the system. This approach ensures that the AR application can work independently of specific devices or applications, making it easier to transfer solutions across different industrial contexts. The standardized architecture promotes flexibility and adaptability, overcoming the limitations imposed by hardware-specific or use-case-specific solutions.
- **Absence of Standardized Development Procedure:** The framework introduces a structured development procedure for AR-HMI (Augmented Reality based Human Machine Interaction) applications, addressing the challenge of the absence of clear guidelines for development. This step-by-step process ensures that industries can follow a clear roadmap to define the specifications of their AR systems and integrate them into existing workflows, reducing the complexities of implementation. The standardized procedure helps to mitigate the uncertainty and inefficiency that often arise from a lack of formal guidelines.
- **Scalability and Interoperability:** The framework addresses scalability and interoperability by defining an intermediary server that facilitates real-time communication between the AR system and the mechatronic system. This server ensures that data synchronization, standardization, and scalability are handled effectively, allowing AR solutions to adapt to different industrial environments without requiring extensive modifications. This scalability and flexibility are essential for enabling AR systems to grow and evolve alongside industrial processes, making them adaptable to a wide range of use cases and settings.

Through these components, the proposed framework effectively addresses the key barriers to AR integration in industrial environments, providing solutions that are scalable, adaptable, and manageable across different industrial contexts.

The proposed framework offers several significant advantages. The structured development procedure provides clarity and consistency, ensuring a simplified process for creating real-time systems for AR-based human-machine interaction in industrial environments. The adaptable object-oriented data model allows data from complex systems to be organised efficiently, supporting both scalability and standardisation across different systems. Finally, the generic AR application model ensures that applications can dynamically generate information based on the underlying data model, enabling seamless integration in different industrial contexts.

The thesis is structured in several sections that systematically explore the research conducted and the results obtained. Initially, the literature review is examined through a critical analysis [2], including an evaluation of the key themes that have emerged, common applications of AR in industry and proven implementations, such as machine monitoring, maintenance, quality control and human-robot

collaboration. This section concludes by defining the identified challenges and future research directions, which motivate the purpose of the research.

The core of the research is represented by the Framework [3], developed to address the challenges identified in the literature review. The section first introduces the motivation behind the creation of the framework and then delves into its architecture, which includes the mechatronic system, AR applications, and the intermediary server. The Design Process is described in detail, with a breakdown of phases focusing on AR application design, mechatronic system information modeling, and server definition. Additionally, an AR application development model based on an information model workflow is introduced, ensuring the solution's flexibility and adaptability. The section concludes with a reflection on the main outcomes derived from the framework's definition.

The framework was validated through a Case Study [4], where the methodology was applied to a Cartesian robot. This section describes the design process applied to the case study, including the AR application design phase, mechatronic system information modeling, and server configuration. The Implementation phase is then analyzed, focusing on the server configuration, mechatronic system controller upgrade, and AR application development. The case study concretely demonstrates how the framework can be applied in an industrial context, with results that validate the proposed methodology.

The Results [5] obtained are described in two main areas: Development in a Simulated Environment, which laid the foundation for creating a digital twin for remote monitoring, and Results from AR Deployment on Magic Leap 2, where the AR application was tested in a real-world context, showcasing the effectiveness of the proposed solution for developing a real-time human-machine interaction system.

Finally, the Conclusion [6] summarizes the overall results of the research, suggesting future directions for the further development and implementation of the framework in various industrial settings. The framework, validated through the case study, demonstrates its ability to address the challenges related to AR integration in mechatronic systems, providing a scalable, adaptable, and manageable approach.

## 2 Literature Review: Analysis and Insights

### 2.1 Introduction

In the opening chapter of this thesis, the main objective was outlined: the development of a framework for designing human-machine interaction systems through augmented reality (AR). This framework serves as a guideline for applying AR tools in interaction with generic mechatronic systems. The subsequent chapters will propose a detailed procedure for designing such systems, addressing various key aspects. These include how to represent the large volumes of data associated with mechatronic systems, how to structure AR applications, and, more broadly, the steps to follow when making these considerations.

The motivation for choosing this topic stems from an in-depth analysis of the state of the art, presented in this chapter, which explores the current use of AR in industrial contexts.

From the analysis conducted, it is evident that augmented reality holds well-established value as an enabling technology in industry. It facilitates more sophisticated, efficient, and socially sustainable interactions in specific processes. This aligns with the human-centric vision of Industry 5.0, where strategically providing additional information to users enhances decision-making and operational efficiency.

Despite the compelling arguments supporting the adoption of AR in industrial contexts, its deployment remains limited due to several challenges. This chapter delves into these issues in detail, presenting a comprehensive analysis of the state-of-the-art AR technology in industry. The conclusions drawn from this thorough preliminary study form the foundation and justification for this thesis work.

The literature review was conducted using Scopus, a citation database developed by Elsevier, by searching for all articles with augmented reality as the primary subject. Filters were then applied, starting with a temporal restriction: given the rapid technological advancements in this field, only articles published from 2018 onward were considered. The application domain served as a second filtering criterion, focusing solely on articles related to industrial applications, while excluding other major fields such as medicine and training. The resulting articles were thoroughly analyzed, and additional relevant works were identified through snowballing, also known as citation chaining, which involves exploring the references and/or citations of selected articles to uncover further pertinent material. This approach was chosen to enrich the results and identify potentially less evident but valuable contributions.

The results of the research revealed the existence of a significant number of literature reviews on the topic, declined to the analysis of different aspects, and numerous research articles, almost all of which, once analysed, turned out to be case studies. On this basis, the state-of-the-art analysis was divided into two main sections.

First, in Section 2.2, the existing literature reviews were investigated and presented in chronological order, including comments on the context of the analysis covered in each article and the key conclusions drawn. From the analysis of these reviews, a comprehensive collection of identified challenges and future research directions was derived. These were categorized into three main areas: industry-related

factors, technology-related factors, and user-related factors. For instance, industry-related factors include the difficulty of integrating AR with existing production systems, technology-related factors highlight limitations in hardware performance, and user-related factors address issues such as usability and training requirements.

Next, the research articles were analyzed in Section 2.3. These articles highlighted the use of augmented reality in various industrial applications and were further divided into subsections based on the topics addressed, such as maintenance, assembly, human-robot collaboration, and quality control.

In conclusion, starting from the shortcomings identified in the literature review, research efforts on various fronts were examined through the available research articles, highlighting a notable lack of studies addressing industry-related challenges. These findings underline the need to develop standards to facilitate the creation of AR applications in industrial contexts. Addressing this open and critical issue is essential to overcome the persistent difficulties encountered in this area.

## **2.2 Analysis of existing literature reviews**

In this initial section, the literature review articles on the application of AR technology in the industrial sector will be analyzed. All relevant works will be presented, with comments on the specific aspects addressed and an analysis of the identified limitations, highlighting the reasons that justify the need for further research in the field.

### **2.2.1 Summary and Commentary on Existing Reviews**

#### **The Impact of Augmented Reality Devices on Operator Performance in Manufacturing Contexts ([1] - 2024)**

The article offers an assessment of the impact of augmented reality technologies in the manufacturing sector by reviewing available case studies. The analysis considers several aspects: the type of device, the application and task to be performed by the operator, and the operator's age and experience with AR and the task to be performed. The conclusions drawn confirm that the use of AR devices reduces task completion time, highlighting that the least effective technology is head mounted display (HMD) devices due to operators' unfamiliarity with them.

#### **Exploring the synergies between collaborative robotics, digital twins, augmentation, and industry 5.0 for smart manufacturing: A state-of-the-art review ([2] - 2024)**

The article explores the synergies between collaborative robotics, digital twins, augmentation technologies such as virtual reality and augmented reality, and artificial intelligence within the framework of Industry 5.0. The primary focus is on how these technologies collectively enhance smart manufacturing by improving human-robot collaboration, increasing efficiency, and promoting sustainable production processes. Wearables play a crucial role in human robot collaboration (HRC) by providing a new level of interaction between humans and robots, making collaboration more effective and safe. VR/AR-based control of robots in HRC can improve the operator's situational awareness, enhance their ability to control the robot, and reduce the need for complex programming or manual input. This approach can increase the efficiency and effectiveness of the manufacturing process while reducing the risk of errors or accidents. Despite the proven advantages of these technologies, their effective deployment faces several challenges, including the complex integration with existing manufacturing processes, the

need for technical experts for development, the initial investment in costs and training, and human factors related to end users, such as resistance to change.

#### **Head-mounted display augmented reality in manufacturing: A systematic review ([3] - 2023)**

The article presents a comprehensive review of the state of Augmented Reality with Head-Mounted Display in the manufacturing sector since 2012. It highlights the significant progress made in various industrial applications such as assembly, maintenance and training. The paper details the technical aspects of HMD AR implementation, including AR display, tracking and recording, context awareness, human-machine interaction, and ergonomic considerations. Although HMD AR technologies have significantly improved and diversified their applications in recent years, further research is needed to put them to more widespread and effective use, such as reliable long-term tracking, ergonomic improvements, and the development of more intuitive interaction methods.

#### **Augmented Reality: Survey ([4] - 2023)**

The article provides an extensive review of Augmented Reality (AR), analyzing its applications, challenges, and future directions across different sectors. AR is recognized for its potential to overlay digital information onto the physical world, creating immersive environments for training, maintenance, design, and quality control in industries like manufacturing. However, its adoption faces several obstacles in industrial settings. Key challenges include the high costs of AR systems, the complexity of the information displayed, and the demand for maintenance. Additionally, the absence of universal standards hampers interoperability across platforms, while cybersecurity and communication concerns persist, particularly in environments that require real-time data processing. AR also struggles with the technical challenge of seamlessly integrating virtual elements with real-world environments, which requires precise rendering and tracking techniques. The need for improvements in user interface design, real-time interactions via 5G and edge computing, and integration with the Internet of Things (IoT) further complicates its application in complex, fast-paced industrial contexts. Despite these hurdles, AR's future is promising as research continues to explore innovative solutions that will enhance its use in industries and society.

#### **Towards Data and Model Interoperability for Industrial Extended Reality in Manufacturing ([5]- 2023)**

The article provides a review of well-known previous studies [6, 7, 8, 9, 10, 11, 12], gathering the research pathways related to the use of extended reality (XR) technologies in the industrial sector. Through this review, the article examines how AR, MR, and VR can be applied in industrial contexts, highlighting the potential benefits in design, production, and maintenance. However, the review identifies a significant weakness that hinders the advancement of XR technologies in industry: the problem of interoperability. The main challenges involve the complexity of integrating different types of data, models, and workflows into XR systems. The variety of data used in manufacturing and the lack of unified standards for models make the widespread adoption of these technologies difficult. Interdisciplinary research and the development of more expressive APIs are suggested as essential steps to overcome these challenges, thereby improving the integration and effectiveness of XR technologies in the industrial context.

### **Mapping the Emergent Trends in Industrial Augmented Reality ([13] - 2023)**

The article provides a comprehensive literature review of augmented reality (AR) trends in the industrial sector from 2018 to 2022. The study identifies ten major topics where AR is most frequently applied: Industry 4.0, artificial intelligence, smart manufacturing, industrial robots, digital twins, assembly, the Internet of Things, visualization, maintenance, and training. The authors conclude that AR has significant potential to enhance industrial processes by improving efficiency, reducing errors, and cutting costs. However, they also highlight several challenges, such as the reliability of AR systems, software support, safety concerns, and overlay accuracy, which currently limit the widespread adoption of AR in industrial environments.

### **Augmented reality-based guidance in product assembly and maintenance/repair perspective: A state of the art review on challenges and opportunities ([14] - 2023 )**

The article provides a review of the application of augmented reality (AR) in product assembly and maintenance/repair activities, particularly in the context of Industry 4.0. The review highlights how AR can significantly improve knowledge transfer during critical steps such as assembly, repair and maintenance, addressing key challenges in these processes. Despite its potential, the article notes that the current application of AR is often limited to experimental settings, while its full integration into industrial environments still faces significant obstacles; including the need for further research to improve virtual interfaces, better understand worker behaviors, and promote collaboration across industries.

### **Towards Industry 5.0: Augmented Reality Assistance Systems for People-Centred Digitalisation and Smart Manufacturing ([15] - 2023)**

This review identifies key topics in human-centric manufacturing, focusing on human operator wellbeing, methods for human-centric manufacturing, and AR assistance systems for smart manufacturing. It highlights the importance of monitoring wellbeing factors, such as musculoskeletal disorders and mental workload, and emphasizes the role of user acceptance in adopting new technologies like AR. The concept of 'Operator 4.0' is discussed, integrating human operators into cyber-physical systems to enhance wellbeing. However, challenges remain in understanding the correlation between wellbeing and performance and in fully leveraging AR's potential.

### **Augmented Reality in Maintenance—History and Perspectives ([16] - 2023)**

This review article explores the integration of augmented reality (AR) with deep learning neural networks for object detection in maintenance tasks. It aims to provide a comprehensive overview of how AR, combined with advanced machine learning techniques, can enhance various maintenance activities such as learning, assistance, remote support, and inspection. The review highlights the significant potential of AR to improve the efficiency and safety of maintenance operations by seamlessly connecting the physical and virtual worlds. Additionally, the article emphasizes the importance of environmental and personal studies to optimize the practical use of these systems. The authors also point out the need for future research in areas like object occlusion and AR hardware to further advance this field.

### **A state-of-the-art survey on Augmented Reality-assisted Digital Twin for futuristic human-centric industry transformation ([17] - 2023)**

The article provides a comprehensive literature review on the integration of augmented reality (AR) with Digital Twin (DT) systems, considering some previous works that focus on singular aspects, [18, 19, 20, 21, 22, 23, 24]. This study examines state-of-the-art research on AR-assisted DT across various engineering lifecycle stages, including design, production, distribution, maintenance, and end-of-life phases. The paper categorizes these improvements into three dimensions: virtual twin, hybrid twin, and cognitive twin. Additionally, four promising future research directions are proposed, focusing on product design, robotics, cyber-physical interaction, and human ergonomics. The article emphasizes that AR not only meets the visualization and interaction needs of DTs but also integrates human intelligence into the system, promoting advanced cognitive capabilities and a human-centric industry transformation.

### **Systematic Literature Review on Augmented Reality-Based Maintenance Applications in Manufacturing Centered on Operator Needs ([25] - 2022)**

This literature review highlights the critical role of Augmented Reality (AR) in enhancing maintenance operations within manufacturing. The analysis reveals that AR is particularly effective in supporting less-experienced operators by providing expert guidance and integrating with Industrial Internet of Things (IIoT) sensors for data collection and ambient intelligence. A key insight is the importance of network communication and ergonomic design in AR systems, with Head-Mounted Displays (HMDs) offering significant advantages over Hand-Held Devices (HHDs) by freeing operators' hands and improving safety. Additionally, the review emphasizes the need for content authoring to reduce operator mental workload by customizing AR content based on the operator's skill level and experience. Future developments should focus on leveraging 5G technology, improving HMD design, and tailoring AR solutions to specific industrial sectors, ultimately contributing to more effective Predictive Health Management (PHM) in manufacturing.

### **Study of Augmented Reality Based Manufacturing for Further Integration of Quality Control 4.0: A Systematic Literature Review ([26] - 2022 )**

The literature review on augmented reality (AR) in manufacturing highlights its increasing role in Industry 4.0, particularly in enhancing quality control. In this field, AR assists operators by superimposing digital information onto the physical environment, enabling real-time, precise measurements and reducing human error. The review identifies three main applications of AR in quality control: virtual Lean tools for error prevention, AR-assisted metrology, and in-line quality control. These applications have proven to streamline operations, improve accuracy, and reduce the mental workload on operators by providing interactive guidance and reducing dependency on traditional tools like paper manuals or video instructions. The article identifies several challenges in applying augmented reality (AR) in quality control. First, while AR technology has matured, there are still issues related to user acceptance. Resistance to adopting new technology can significantly impact its long-term effectiveness in manufacturing settings. Another challenge lies in the complexity of integrating AR with existing quality control processes, which often require precise and repetitive tasks. AR systems need to provide accurate guidance to operators without increasing their mental workload. Additionally, there is a need

for robust evaluation mechanisms to ensure the usability and effectiveness of AR solutions in different environments. This includes managing the cost, time, and accuracy of the systems, which is essential for long-term benefits. AR's integration with intelligent technologies, such as artificial intelligence (AI) and data-driven solutions, is still in the early stages and presents a hurdle for real-time application in quality control.

### **Augmented reality situated visualization in decision-making ([27] - 2022)**

This article reviews the integration of Situated Visualization (SV) with Augmented Reality (AR) to enhance Decision Support Systems (DSS). It explores how AR, by providing contextual and relevant data visualizations, can improve decision-making processes, making them more efficient and effective. The paper discusses the benefits, challenges, and opportunities of using AR for SV in various applications, such as construction, architectural design, industrial maintenance, and safety management. It highlights various challenges that affect the implementation and effectiveness of AR in industrial contexts. These challenges include the complexity of visual coherence, temporal accuracy, visual interference, handling data overload, and ensuring seamless integration of digital content with the real-world environment. Additionally, AR's egocentric viewpoint and the dynamics of real-world situations present difficulties. The article also emphasizes the need to rethink design, methods, and evaluation approaches for AR-based decision support systems (DSS), especially in industries like construction, architectural design, and industrial maintenance. Although the potential of AR to enhance decision-making is recognized, its slow adoption is attributed to technical, methodological, and conceptual challenges, as well as the necessity for more research in fields such as simulation, AI, and human-computer interaction to improve its usability and effectiveness.

### **Augmented Reality for Human–Robot Collaboration and Cooperation in Industrial Applications: A Systematic Literature Review ([28] - 2022)**

This literature review was conducted to analyze the use of augmented reality (AR) in the context of human-robot collaboration and cooperation (HRC) in industry by analyzing publications from 2016 to 2021. The primary objectives were to identify the predominant AR visualization technologies, key application domains, and the current status of AR applications in this industrial context. Most studies indicate that AR assistance improves operational efficiency in human-robot collaboration tasks. In particular, AR systems tend to reduce task completion time, improve ergonomics, and increase user satisfaction, although effectiveness varies among different AR methods, such as Head-Mounted Displays (HMDs) and projectors. These systems generally reduce perceived operator workload and improve performance, although the impact on concentration levels and user confidence may differ. Future research should focus on real-world industrial testing, refining AR interfaces, and setting standards for usability and safety of AR systems in human-robot collaboration environments.

### **A survey of industrial augmented reality ([12] - 2020)**

The review focuses on the application of Augmented Reality (AR) in industrial environments, particularly within the framework of Industry 4.0. It explores how AR is being utilized to enhance various industrial processes, including manufacturing, quality control, maintenance, and training. The study provides an in-depth analysis of the current state of AR technology, examining its benefits, limitations,

and potential for future development. The review focuses on the application of Augmented Reality (AR) in industrial environments, particularly within the framework of Industry 4.0. It explores how AR is being utilized to enhance various industrial processes, including manufacturing, quality control, maintenance, and training. The study provides an in-depth analysis of the current state of AR technology, examining its benefits, limitations, and potential for future development.

### **Augmented reality in support of Industry 4.0—Implementation challenges and success factors ([29] - 2019)**

The article focuses on understanding the challenges and success factors in implementing Augmented Reality (AR) within Industry 4.0. The study combines quantitative and qualitative approaches, using the Technology, Organization, and Environment (TOE) framework to assess the success of AR deployment in industrial settings. While AR is recognized for enhancing efficiency and providing real-time, hands-free information, the research reveals that its implementation in industry faces several challenges. The reasons for the limited adoption of AR in industrial settings include the complex integration of AR with existing systems, technical challenges such as hardware readiness and compatibility, and issues related to user acceptance. Ergonomics, the usability of AR interfaces, and visibility of information are significant concerns, as users often experience discomfort or inefficiency, particularly when using head-mounted displays. Organizational barriers, including the need for employee training and adaptation of business processes, as well as the technological immaturity of AR in certain areas, further impede widespread adoption. Additionally, while external support could aid in AR implementation, it was found to have an insignificant impact in this study.

#### **2.2.2 Key themes**

The following commentary synthesizes the key themes and organizes them based on the recurring topics identified in the reviews.

**Synergies and Applications** Several articles focus on the synergies between AR and complementary technologies, such as collaborative robotics, digital twins, and artificial intelligence, in the context of Industry 5.0. For example, [2] highlights the potential of AR in enhancing human-robot collaboration and situational awareness while addressing challenges like integration with existing processes and user resistance. Similarly, [17] explores the integration of AR with digital twins, categorizing advancements into virtual, hybrid, and cognitive twins. These works emphasize the importance of interoperability and rendering precision, which are critical for advancing AR applications.

**Device-Specific Analyses** The role of specific devices, particularly Head-Mounted Displays (HMDs), is explored in depth by [3] and [25]. [3] highlights the technical progress in AR-HMDs for applications like assembly and maintenance but identifies persistent challenges such as ergonomic improvements and intuitive interaction methods. [25], on the other hand, underscores the benefits of HMDs in reducing operator workload while calling for advancements in network communication and 5G integration. These findings highlight the importance of usability, ergonomics, and computational performance in AR devices.

**Industrial Integration and Interoperability** Interoperability is a recurring theme in the reviews, particularly in [5], which examines the integration of diverse data and workflows into XR systems. The lack of universal standards is a significant barrier, as noted by [4], [12], and [29], who collectively stress the need for standardized frameworks to overcome integration challenges. These issues are central to enabling seamless adoption of AR technologies in industrial contexts.

**Application-Specific Reviews** Several articles delve into specific applications of AR in industrial settings. [14] and [16] focus on maintenance and repair, highlighting AR’s potential to improve knowledge transfer and operational efficiency. Similarly, [26] and [28] analyze AR’s role in quality control and human-robot collaboration, respectively. These studies reveal gaps in the adoption of AR due to challenges such as user training, learning curves, and resistance to change.

**Human-Centric Perspectives** Finally, [15] and [27] emphasize the human-centric aspects of AR in industrial contexts. [15] discusses the integration of user wellbeing into AR-assisted systems, addressing challenges like musculoskeletal disorders and mental workload. [27] explores the benefits of AR in decision support systems, identifying visual coherence and data overload as key obstacles. These insights underline the need to enhance user acceptance and interaction methods to fully realize AR’s potential.

**2.2.3 Identified challenges and future research directions**

The synthesis of these reviews reveals recurring challenges and opportunities in the adoption of AR technologies in industrial environments. These insights provide a foundation for categorizing the identified shortcomings and future research directions in technology-, user-, and industry-related factors. Figure 1 depicts this concept.

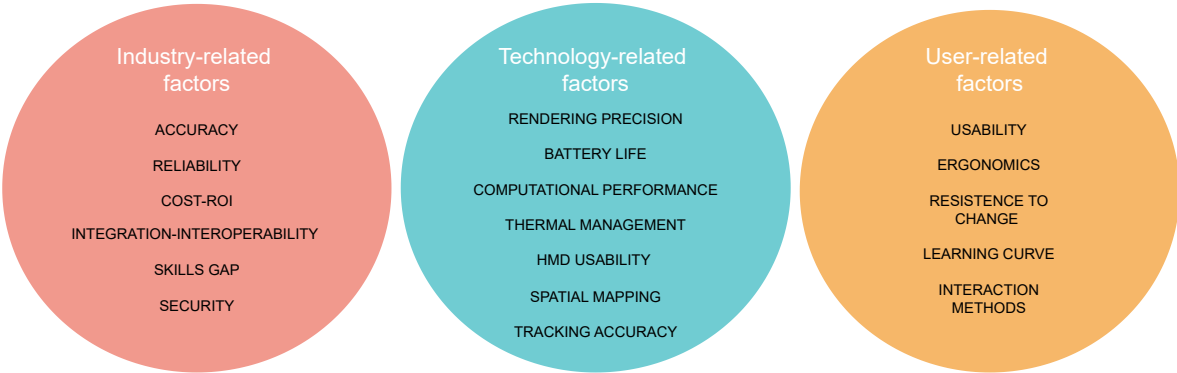


Figure 1: Research openings for the use of AR in industry

**Industry-related Factors**

These factors are linked to the broader operational and economic aspects of deploying augmented reality (AR) in industrial settings. It is worth noting that while accuracy and reliability may initially appear to be technology-related, they are classified here as industrial factors to emphasize their impact on the practical use of AR solutions in the industrial context, separating the technological causes from the operational effects.

- **Accuracy:** Industrial applications demand high precision to ensure that AR systems provide reliable and actionable insights, especially in manufacturing and maintenance.

- **Reliability:** Consistent performance over time is crucial to gain trust in AR solutions, as downtime or errors can lead to significant operational losses.
- **Cost-ROI:** While AR systems can offer substantial benefits, their adoption hinges on demonstrating a clear return on investment to justify upfront costs.
- **Integration-Interoperability:** Seamless integration with existing systems and equipment remains a challenge, as industrial environments often involve heterogeneous technologies.
- **Skills Gap:** Beyond using AR applications, there is a need for personnel capable of developing and customizing AR solutions to fit specific industrial requirements.
- **Security:** Ensuring data integrity and protecting sensitive information in AR systems is paramount, especially when dealing with industrial processes and intellectual property.

### Technology-related Factors

These address the performance and limitations of AR devices and platforms from a technical perspective:

- **Rendering Precision:** High-quality visualizations are critical for industrial AR applications to convey accurate and clear information.
- **Battery Life:** Extended usage in industrial scenarios requires AR devices to have long-lasting power sources, especially for field operations.
- **Computational Performance:** Real-time data processing is essential to ensure smooth operation and immediate feedback in dynamic environments.
- **Thermal Management:** Prolonged use in industrial settings can lead to overheating, impacting both device performance and user safety.
- **HMD Usability:** Head-mounted displays need to balance functionality with comfort to support prolonged use without causing fatigue.
- **Spatial Mapping:** Accurate environmental mapping is vital for AR applications to anchor virtual content effectively in the real world.
- **Tracking Accuracy:** Reliable tracking systems ensure that virtual overlays align precisely with real-world objects, which is crucial for tasks like assembly or maintenance.

### User-related Factors

These pertain to the human element of AR adoption, focusing on usability, acceptance, and user experience:

- **Usability:** AR interfaces must be intuitive and user-friendly to ensure accessibility across a diverse workforce.
- **Ergonomics:** Designing devices that are comfortable to wear and use for extended periods is essential to prevent fatigue or injury.
- **Resistance to Change:** Overcoming skepticism or reluctance from workers requires addressing concerns and demonstrating clear benefits.

- **Learning Curve:** AR systems should aim to minimize the effort and time needed for users to become proficient.
- **Interaction Methods:** Effective and intuitive interaction mechanisms, such as gesture control or voice commands, enhance the overall user experience and adoption rates.

This categorization into three areas highlights the main open research challenges that currently hinder the widespread adoption of augmented reality in industrial contexts. The next section will present research efforts related to AR in industry, and the conclusions will discuss how the field is advancing in addressing the various challenges identified.

## 2.3 Common applications and proven implementations

As introduced earlier, the state-of-the-art analysis has highlighted the presence of numerous literature reviews, as well as various research contributions with a more practical focus. Many of these works provide practical solutions to improve the execution of specific operations in the industrial sector, such as Machine and Process Monitoring, Maintenance, and Quality Control. The following sections present these contributions, categorized by the specific topics addressed.

### 2.3.1 Usability and Interfacing in AR for Industrial Applications

Several studies focus on enhancing user interaction with AR systems through hands-on control and gesture recognition. One such example is the work of Blankemeyer (2024) [30], where AR is employed as a communication bridge between operators and robots, allowing the operator to demonstrate tasks through hand gestures. This interaction style enables those with process knowledge but no programming skills to effectively control robotic systems. Similarly, Kolla (2023) [31] explored different AR interaction modalities in assembly tasks, including touch, voice, and gesture, highlighting the strengths and weaknesses of each in industrial settings. For example, while touch was intuitive, it posed limitations in environments where operators wore gloves. Gesture control offered promise, but computational demands and complex hand movements required further refinement.

These approaches point towards the potential for AR to streamline interaction between human operators and automated systems, although challenges remain regarding practical usability in specific industrial conditions.

Another key theme is the impact of AR on ergonomics and user experience. Pereira (2023) [32] discusses how AR technologies can mitigate physical and cognitive strain in industrial logistics environments. The study finds that workers view AR technologies like exoskeletons and wearable warning devices (WWD) as beneficial for reducing physical exertion, though cognitive workload tools (e.g., AAR and HMD) were considered less critical.

On a broader scale, Alexandre (2024) [33] analyzed the user experience (UX) of AR in Industry 5.0 applications, noting that different user profiles (general users vs. human factors professionals) respond differently to AR systems. The system was seen positively overall, but background expertise shaped perceptions of AR's potential. Margolis (2024) [34] further refined these AR systems by enhancing visualization and interaction features, improving user feedback and effectiveness, albeit with some persistent issues like font size.

Both studies reflect the ongoing balance between user expectations and the technological maturity of AR systems, emphasizing the importance of ergonomics and intuitive design in achieving user acceptance.

A shift towards visual-heavy documentation is apparent in industrial AR applications. Gattullo (2019) [35] introduces a novel approach to technical documentation, where AR interfaces reduce reliance on text by emphasizing visual communication. Although some users initially perceived a lack of content due to minimal text, the methodology was generally well-received, as information was found to be clearer and more accessible. This change marks a broader trend towards prioritizing visual aids in AR, which improve user understanding and interaction during maintenance and operational tasks.

Similarly, Chen (2024) [36] highlights how AR-based visualization can improve decision-making by offering intuitive, multi-modal perception and interaction environments. The system's usability is enhanced through real-time user data acquisition, such as eye-tracking, allowing for more immersive and effective interactions. The article stresses the potential of AR systems in facilitating complex decision-making by enhancing user perception and memory retention.

An important development is the simplification of AR programming for non-expert users. Li (2024) [37] introduces a graphical programming platform designed to allow users to create interactive AR interfaces without extensive coding knowledge. By dividing AR development into front-end and back-end processes, users can manipulate AR models once specific images are recognized, using a simplified graphical interface. This platform lowers the barrier for AR development in industrial settings, making it more accessible to users without technical programming skills, though the absence of in-built modeling features remains a limitation.

Rendering techniques for AR interfaces play a crucial role in improving user interaction and task efficiency. Fortuna (2024) [38] evaluates the impact of different rendering techniques on industrial inspection tasks. The study identifies that the Outline (RTB) and Semi-transparent (RTA) rendering styles are particularly effective in enhancing the clarity and usability of AR interfaces for these tasks, suggesting that specific rendering methods can optimize user performance and interaction in industrial applications.

Finally, Marino (2024) [39] explores a context-aware recoloring approach that adjusts the visibility of virtual objects in real-world environments, aimed at improving AR-assisted maintenance tasks. This method enhances usability by allowing users to differentiate between virtual and physical objects more easily, addressing a common usability challenge in AR interfaces. Future experiments will focus on evaluating the approach with end-users in industrial settings, expanding on the system's potential to improve practical applications of AR in complex work environments.

In summary, these studies collectively highlight the importance of user-centric design in AR for industrial applications, focusing on hands-on interaction, ergonomic benefits, enhanced visual documentation, and the role of specialized rendering techniques in improving the overall usability of AR systems in the workplace.

### **2.3.2 Machine and Process Monitoring in Industrial Applications using AR**

The use of augmented reality (AR) in machine monitoring has increasingly become an integral part of modern industrial processes, providing enhanced visualization and interaction with real-time data. Various studies and projects have explored how AR can be effectively integrated with other technolo-

gies to improve operational efficiency and decision-making in manufacturing environments.

Maio's research (2024) [40] investigates the application of Microsoft HoloLens 2 in monitoring assembly line operations, specifically in a boiler assembly line with sixteen workstations. Participants used the HoloLens to track metrics like bottlenecks and delays while performing their tasks. The study utilized a Human-Centered Design (HCD) methodology to identify areas for improvement and define specific requirements. The comparison of AR with a web-based tablet application revealed that while AR was less efficient for general task management, it excelled in tasks requiring rapid decision-making and quick problem resolution due to its ability to visually associate information with specific workstations. However, user preference for the familiar interface of the web application highlights the need for further development in AR usability.

Eber's work (2023) [41] contributes to the understanding of AR's role in enhancing situational awareness and decision-making in manufacturing contexts. The study employed sensors to monitor various system parameters within a learning factory and visualized this real-time data through AR. Utilizing tools such as CP Factory machinery, PTC ThingWorx, Vuforia Studio, and Microsoft HoloLens 2, the approach demonstrated the feasibility of integrating standard market software for effective AR-based monitoring. This solution, accessible to SMEs due to its minimal requirement for specialized knowledge, showcases AR's potential to improve real-time data visualization and machine monitoring.

Patel's project (2024) [42] involves the S-NODE device, which monitors physical parameters to predict maintenance needs. Data from the S-NODE, stored in an MS SQL database, is used to train machine learning models. The AR application, developed with Unity Editor and Vuforia, tracks the S-NODE device using QR codes to display real-time data according to device type and parameters. This project highlights the application of AR in monitoring and predicting machine maintenance, integrating AR with IoT data for enhanced operational insights.

Jakl's study (2018) [43] examines two AR use cases designed for Industry 4.0 environments. The "Real-Time Machine Data Overlay" use case involves placing holographic dashboards in the real world to visualize machine data in real-time, while the "Web-Based AR Remote Support" use case enables remote experts to provide assistance through AR annotations. Both prototypes use Microsoft HoloLens and OPC UA for data retrieval, demonstrating AR's capability to support real-time visualization and remote collaboration.

Gramberg's case study (2023) [44] focuses on an AR dashboard for sensor data visualization in a smart manufacturing environment. Using Vuforia and MQTT protocol, the application allows interaction with a test bench through AR interfaces on HoloLens, illustrating how AR can facilitate real-time monitoring of machine parameters.

Filip's application (2023) [45], developed with Unity and Vuforia, provides AR-based dashboards for monitoring production and sorting lines. This application allows users to interact with and view detailed information about machine operations through AR, enhancing monitoring and control capabilities.

Novaliendry's research (2024) [46] explores an industrial monitoring system combining IoT and AR to

create a digital twin prototype of a conveyor machine. By integrating real-time data with AR visualizations, the system offers accurate monitoring and representation of the physical machine, showcasing how AR and IoT can enhance industrial monitoring processes.

Schmitt's research (2023) [47] focuses on AR-based energy monitoring, comparing visualization tools like Power BI and Unity. The study develops data pipelines to transform real-time energy data into AR-based dashboards, using Microsoft HoloLens 2 and Apple iPad Pro. This application of AR extends beyond machine monitoring to include energy consumption tracking, demonstrating AR's versatility in industrial applications.

Stark's study (2020) [48] introduces a dynamic GUI generation system for mechatronic systems, leveraging AR and IoT technologies. By recognizing devices through 3D object recognition and matching them with digital twins stored in the cloud, the system generates a real-time GUI for device control, enhancing interaction and monitoring capabilities.

These studies collectively illustrate the diverse applications of AR in machine monitoring, from predictive maintenance and real-time data visualization to energy monitoring and remote support. By integrating AR with IoT and other technologies, these solutions enhance operational efficiency, decision-making, and situational awareness in industrial environments, paving the way for more advanced and responsive manufacturing processes.

### **2.3.3 AR-assisted industrial maintenance**

Augmented reality (AR) has emerged as an important technological advancement in industrial maintenance, offering innovative solutions to increase efficiency, reduce downtime and improve task accuracy. Traditional maintenance methods often involve manual inspections and paper-based procedures, which can be inefficient and error-prone. AR addresses these limitations by overlaying the physical environment with digital information, such as equipment schematics, operational data, and maintenance instructions. This literature review examines the integration of AR into maintenance practices, highlighting its role in transforming how industrial maintenance is performed.

Giliyana (2024) [49] introduces a testbed designed to train companies on key Industry 4.0 technologies for developing smart maintenance solutions. Surveys and interviews reveal that the most common challenge in implementing these technologies is a lack of knowledge. The testbed features a dashboard displaying a QR code for digitized maintenance instructions, which, when scanned with AR glasses, provides a step-by-step guide with accompanying figures and text.

Liu's research (2022) [50] focuses on predictive maintenance in machine tools, combining AR with IoT technologies. The system acquires machine specifications and sensor data, which is processed using deep learning techniques to predict faults. If a fault is predicted, the AR interface provides guidance through marker and object detection, displaying maintenance instructions and connecting users with remote experts if needed. This system, developed using Unity 3D and Vuforia, illustrates how AR can enhance maintenance processes by providing real-time insights and support.

Peansupap's research (2024) [51] discusses the development of an Augmented Reality (AR) prototype system designed to assist in decision-making for selecting and preparing suitable temporary structures

in building maintenance. The AR system aids maintenance personnel by providing safety checklists and a reporting system to ensure the safe installation of these structures. Internal and external validations were conducted to assess the system's accuracy, showing that it effectively supports maintenance tasks. However, challenges were identified, particularly related to the markerless AR system's sensitivity to environmental lighting, ground textures, and camera orientation, which affected object tracking and caused floating of virtual objects. The article suggests future improvements to the system, including enhanced tracking capabilities and further testing of the safety checklist module, which was limited in this research due to the unavailability of mobile scaffolding.

In [52], Simon (2023) presents a remote maintenance system using a mobile robot controlled via a Raspberry Pi, allowing remote navigation within a facility. The augmented reality (AR) application enables users to visualize 3D models of objects when the mobile robot's camera captures them. The application is developed in Unity and utilizes Vuforia for target recognition. The primary focus of the research is on testing the system's ability to detect objects, specifically examining the influence of distance and the number of markers associated with each object. The study emphasizes the development of an AR application for remote maintenance with particular attention to object detection and target recognition.

Another paper focused on remote maintenance is [53], where two AR-based case studies focusing on remote maintenance of industrial manipulators are described. The first study involves an on-site AR system using see-through glasses to guide a technician during a local repair task. The second study explores a remote approach where an expert can visualize and inspect the machine using a tablet and a high-precision tracking system, allowing them to virtually walk around the machine and better understand the production process. Both studies were evaluated with participants, yielding mixed results and highlighting the need for further research to improve interaction methods and the usability of these AR solutions.

A different paper focusing on assisted maintenance is Frandsen's (2023) [54]. The paper presents a system based on augmented reality to guide the user in positioning to capture images of the scene to be analyzed. Positioning is achieved using a 3D model of the Festo testbed, and Vuforia for tracking. The prototype is developed for handheld devices.

In [55] instead, Valentini presents a user experience (UX)-driven methodology aimed at designing user-centric augmented reality (AR) applications for complex maintenance procedures in industrial settings. The methodology was applied to a real industrial case involving the management of CNC machines in a tractor component manufacturing plant. The researchers developed and tested a smartphone-based AR application with users, yielding satisfactory results that highlighted the potential of AR in improving industrial maintenance operations. To design the AR interface, the initial prototyping was carried out using Adobe XD to create the graphical user interface (GUI) and define interaction behaviors. The full AR application prototype was then developed using the Unity 3D platform. Image Targets were integrated using the Vuforia Target Manager, linking the virtual elements to their respective QR code targets.

Another application for AR-assisted maintenance is presented in [56]. The prototype AR application developed by Cachada (2019) specifically aims to improve maintenance operations in an automotive

metal stamping production unit. The application is designed to provide real-time training and guidance to personnel during maintenance tasks, addressing a critical need for efficiency and precision in industrial settings. The application, developed using the Unity platform, employs Vuforia’s image tracking technology to initiate maintenance procedures only when the user is at the correct starting point, as identified by specific markers. This marker-based approach ensures that maintenance tasks are performed in the correct sequence and location, reducing the risk of errors. The system also incorporates data collection through a MySQL database, where information about the user, procedures, and task timings is stored. They use two hardware mobile devices: a head-mounted display, Microsoft HoloLens, and a mobile Android Tablet Lenovo 10”.

Aromaa (2018) [57] makes instead a contribution to the field of augmented reality in industrial maintenance, particularly in addressing the challenges associated with the transfer of tacit knowledge—knowledge that experienced maintenance personnel possess but is often not formally documented. The research highlights how AR and advanced communication technologies can bridge this gap by providing easy access to contextually relevant information in various maintenance scenarios. The developed AR system, implemented on an Android smartphone, integrates both marker-based and planar image-based tracking using the VTT’s ALVAR SDK. This system is designed to support maintenance tasks by displaying real-time status information of electrical systems, such as condition and fault codes, while also enabling social media-like features for sharing notes and images among service personnel. The social media functionality is supported by a proprietary server that collects and shares user-generated notes, comments, and observations, facilitating collaborative knowledge sharing among maintenance teams. This approach is particularly beneficial in scenarios where up-to-date information is crucial but may not always be readily available.

In [58], Angelopoulos (2022) explores the development and application of an intelligent product service system (PSS) designed for adaptive maintenance of manufacturing equipment, incorporating Industry 4.0 technologies such as the Internet of Things (IoT) and augmented reality (AR). The manuscript’s contribution extends to providing an optimization algorithm to adapt stakeholders’ schedules based on the energy supplier’s forecast. The system aims to optimize maintenance processes by also leveraging AR to provide real-time interactive instructions and visualizations for assembly and disassembly activities. The AR component is driven by an algorithm that interprets CAD data to generate accurate maintenance instructions, improving user understanding and efficiency.

Tobiskova (2024) [59] evaluates an augmented reality (AR) prototype designed to improve user guidance during the industrial tool change procedure, focusing on the task of calibrating tool measurements. To address these challenges, the AR assistance tool was developed and evaluated for its usability and the cognitive load it places on users during tool calibration tasks on the production floor. The study utilized the HoloLens 2 as the hardware platform for the AR application. Developed using the Unity game engine, the application integrated the Mixed Reality Toolkit (MRTK) and Vuforia packages to enhance its functionality. MRTK was used to configure the project settings for HoloLens 2, providing materials, menus, and prefab objects specifically designed for AR applications. Vuforia facilitated image and object recognition, allowing the HoloLens 2’s camera to identify pre-configured image targets.

A different idea is presented by Laviola (2024) [60], who presents a user study evaluating the effectiveness of a virtual mirror compared to a real commercial mirror for maintenance tasks involving

occluded components. The study aims to determine whether the virtual mirror can perform as effectively as a real mirror without compromising user performance in terms of task completion time, accuracy, or cognitive load. The results show no statistically significant differences between the two types of mirrors, indicating that the virtual mirror can replace the physical mirror with similar performance. The AR interface for this study was implemented with Microsoft HoloLens 2, and software development was done with Unity 3D Engine, incorporating Mixed Reality Toolkit (MRTK) packages for interaction management and Vuforia Engine for image tracking. An image target was placed inside the right panel of the machine, allowing the system to accurately position and track AR content.

Another paper that focuses on the occlusion issue is [61]. Indeed, the paper presents the development and evaluation of an augmented reality (AR)-assisted method to guide visual assembly and maintenance activities of avionics equipment, particularly in areas with limited visibility. The proposed system uses an advanced pose tracking method based on RGBD data to superimpose assembly instructions on real objects. This AR method improves the efficiency, usability, and learning of tasks by reducing the pressure of user interaction and providing intuitive guidance through 3D model animations and voice instructions.

**Stages development in assisted maintenance** A separate branch of research in Ar-assisted maintenance concerns the creation of the steps to be followed by operators during the intervention.

Specifically, this topic is addressed in [62]. Indeed, this article shows how information can be derived from the 3D model of an object to be visualized during AR-assisted maintenance. To validate the proposal, a mobile application was developed to guide operators during maintenance work on the spindle drive of EMCO Mill 55. Starting from the 3D model of the object on which the operator needs to work, through a series of steps it is possible to obtain a series of 3D models of the various maintenance steps suitable to be visualized in AR through a mobile device. Using Vuforia and tags, each stage can be correctly superimposed on the operator's view.

An alternative is proposed by Palmarini (2023) in [63]. FARA is an innovative geometry-based method designed to streamline the creation of AR content for maintenance tasks. It operates under two primary assumptions: first, that machines can recognize objects within their environment, and second, that CAD models are available for all objects involved in maintenance procedures. Unlike other AR systems that may be time-consuming and require manual programming, FARA simplifies the process by automatically generating AR maintenance procedures with minimal input from the user. The maintainer only needs to focus on performing the maintenance task itself, while FARA takes care of recording the spatial transformations of objects within the field of view (FOV) and building the AR procedure accordingly.

An adaptive proposal, which therefore targets specifications to operator characteristics is presented in [64]. The article explores indeed the development of Augmented Reality-based Adaptive Assistance Systems (ARAAS) designed to deliver information that is both detailed and appropriately tailored to the user's needs. The authors propose a methodology for creating such systems, focusing on adaptive assistance in industrial maintenance, specifically during the overhaul of landing gear. This system adapts to the operator's performance, preferences, and cognitive workload by adjusting elements of the assistance procedure without compromising task feasibility. The adaptation strategy is based on

linking operator data to specific responses from the system. Feedback from inspectors indicates that the ARAAS effectively provides customized support, though the system's long-term sustainability and engagement levels require further study. Initial results suggest that adaptive systems like ARAAS may keep operators more engaged over extended periods compared to non-adaptive approaches.

Klimant, in [65], discusses the challenges associated with the creation process of AR documentation for machine tool maintenance and explores new concepts to address these issues, deepening an earlier research by the same authors [66]. It begins by examining the current state of creating AR content in production environments, revealing that the typical processes rely heavily on specialized authoring systems. These systems require a significant amount of manual effort from maintenance planners, making the process labor-intensive and inefficient. To address this problem, the article analyzes the necessary maintenance data, creation processes, and systems to form the foundation for new documentation concepts aimed at reducing this manual workload. A particular focus is placed on the use of CAD applications as a starting point for the documentation process. Following this analysis, the article develops main concepts for documentation systems and processes tailored to different company requirements. It examines three documentation methods: technical documentation systems, spreadsheet applications, and Product Data Management (PDM) systems. Each of these systems offers a graphical user interface for maintenance planners and organizes the created data in a structured format that is suitable for AR maintenance applications. Finally, the article concludes with a verification of the developed concepts through sample implementations of their components, demonstrating the practical application of the proposed documentation methods.

Another contribution is presented in [67]. The study developed the ARAUM (Augmented Reality Authoring for Maintenance) prototype. ARAUM is designed to automate the creation of AR content in real-time, allowing maintenance experts to focus on specifying the information to be displayed, its format, and its sequence. The prototype consists of two main platforms: the Authoring Platform and the Application Platform. The Authoring Platform, is a desktop interface that enables maintenance experts to interact with 'Information Frameworks' used to generate AR content. These frameworks provide the necessary data for creating contextualized or animated content based on the inputs defined by the experts. This allows for automated content creation without requiring deep AR expertise from the users. The Application Platform Modules, generate AR content automatically based on the inputs from the Authoring Platform. These modules are responsible for contextualizing and rendering the maintenance information, ensuring it is displayed in the correct sequence. The system is designed to accommodate both novice and experienced technicians, offering validation features for novices and flexibility for more experienced users to adapt or create new steps. The Application Platform was developed using Unity 3D for Android handheld devices and the Vuforia SDK for markerless tracking. This choice of software supports both online and offline applications, providing a robust framework for AR content creation and deployment. The components of ARAUM are further demonstrated using a gear box representation as an example for this research.

### **2.3.4 Augmented Reality in Industrial Quality Control**

Augmented Reality (AR) has demonstrated significant potential in enhancing industrial quality control by improving task performance, reducing mental workload, and offering real-time, interactive inspection capabilities. Several studies have explored various AR applications in quality control, each contributing to the understanding of AR's impact on quality management processes.

One of the studies in this area is Seeliger’s investigation (2023) [68] into the development of an AR system for head-mounted displays (HMDs) like Microsoft’s HoloLens 2, aimed at assisting in industrial quality inspection. The system uses visual cues that directly highlight defects on physical objects, synchronizing 3D models and 2D images with real-world products as they move or rotate. This synchronization allows operators to visually inspect products more effectively, as the AR system overlays defects in real-time onto the physical object. Seeliger’s research found that the AR-assisted quality inspection system significantly improved task performance, particularly in more complex tasks, when compared to traditional methods such as screen-based assistance or no assistance at all. Additionally, the system was noted for reducing mental workload among users, while receiving positive feedback regarding user experience. This is a particularly valuable finding for industrial settings where efficiency and accuracy in quality inspection are paramount. By integrating AR into the inspection process, the study suggests that manufacturers can not only boost the precision of defect detection but also enhance the overall user experience, potentially leading to faster adoption of AR technologies in industrial contexts. Furthermore, the focus on human factors within this research highlights how AR can not only improve performance but also create a more intuitive and less mentally demanding work environment. The positive user feedback suggests that AR interfaces may play a key role in the future of quality control, especially as systems become more refined and integrated with industrial processes.

Nowak’s work (2021) [69] on AR for industrial process monitoring complements Seeliger’s findings, but it focuses more on the visualization of data for quality control. This research developed a prototype application that leverages augmented reality to visualize Industrial Process Tomography (IPT) data. IPT is a non-intrusive imaging technique used for monitoring industrial processes and ensuring product quality. In this case, AR is used to allow operators to visualize and interact with IPT data through the HoloLens headset. The AR interface provides a more intuitive way of understanding complex data streams, helping operators monitor processes in real-time and make adjustments based on visualized data patterns. By using AR for quality control data visualization, Nowak’s study underscores how AR can bridge the gap between complex data and actionable insights, making industrial quality monitoring more effective and accessible.

In Barbieri’s study (2024) [70], AR is applied to real-time inspection activities, placing a strong emphasis on user-centered design to ensure the tool’s acceptance and efficacy in industrial environments. The marker-based AR tool, designed for Android smartphones and tablets, facilitates product inspection by enabling users to identify discrepancies between design and assembly. Through 3D annotations in the augmented environment, workers can communicate discrepancies directly with technical teams, enhancing documentation and collaboration during the inspection process. This approach not only improves inspection accuracy but also fosters better communication between different teams within the manufacturing environment.

These studies collectively illustrate the application of AR in quality control, highlighting its role in improving inspection efficiency, accuracy, and user experience. By integrating AR into quality management practices, industrial processes can benefit from enhanced real-time data visualization, interactive inspection capabilities, and streamlined communication.

### 2.3.5 Human-Robot Collaboration and AR

The integration of augmented reality (AR) in industrial settings has opened new avenues for enhancing human-robot collaboration, particularly in the assembly processes. A study by Yanzhang [71] focuses on using AR to improve collaborative assembly tasks between humans and robots. The system they developed relies on a headset to guide the human operator through the assembly of parts while the robot manages tasks like sorting and arranging screws. The collaborative setup was evaluated through tests with ten participants, who affirmed the effectiveness of the approach in improving productivity and streamlining interactions between humans and robots during assembly. This research underscores the practical benefits of AR in optimizing the synchronization of human and robotic efforts in industrial tasks.

Building on similar principles, Makris (2016) [72] presents a more comprehensive AR tool designed to support operators in hybrid environments where humans and robots work together. This system goes beyond task execution, providing real-time production and process information directly into the operator's field of view. It not only facilitates task performance but also increases the operator's sense of safety, an essential aspect when working near large industrial robots. The tool immerses users in a workspace where safety mechanisms and production data are continuously visible, improving efficiency while enhancing situational awareness. By utilizing Unity3D for 3D model management and Qualcomm's Vuforia library for marker recognition, this AR tool efficiently integrates immersive instructions with real-time operational data, making the collaborative environment safer and more intuitive.

Another significant aspect of AR in human-robot collaboration is its role in enhancing worker safety and productivity. The application developed by Michalos (2016) [73] offers an AR-based system running on Android tablets, aimed at ensuring safe interaction between humans and robots while simultaneously improving productivity. This tool visualizes safe working zones, provides production data, and issues visual and auditory alerts, which are critical for preventing accidents and optimizing workflow in industrial settings. Early experiments highlighted how easily the system could be deployed using simple setups like PCs or handheld tablets, making it accessible for various industrial environments. However, challenges remain, particularly in integrating the AR system with legacy systems to automatically retrieve CAD data and production information. Future improvements aim to make data retrieval more intuitive and streamline interactions between the operator and the system, tying specific production phases to the relevant AR functionalities.

In more recent developments, as explored by Schmidt (2022) [74], AR is also being used to facilitate more efficient data exchange between humans and robots. Their study focuses on creating a visual interface for users working with robotic cells, specifically an ABB robotic cell, even in simulated environments. Leveraging Microsoft's HoloLens 2, this AR application allows operators to visualize critical information about the robotic cell's operations. It employs MQTT protocols and JSON as the data interchange format, enabling real-time data flow between the AR interface and the robotic system. This seamless integration of data exchange improves the user's understanding of robot movements and statuses, further enhancing the collaborative potential between humans and robots in industrial settings.

Another example of augmented reality in human-robot collaboration is illustrated in [75]. Chacko

(2019) introduces an innovative AR interaction method that enables a robot to handle unfamiliar physical objects in a shared workspace. A mobile AR application was developed to show in real time, the location, orientation and size of random objects within the robot’s workspace. The approach uses a virtual AR element superimposed on the live view of the real object, which can be used by the user to get information about it and to command its grasp.

The use of augmented reality as a tool for more effective data exchange is also used in the case of mobile robots. In [76], an AR application is presented that allows the user to visualize the location of a multitude of mobile robots, and the movements they will make. This type of application allows the user to be aware of changes in the system they are currently in, consequently feeling safer.

Another example is [77] where operators can control robots in real time through devices like Microsoft HoloLens. With AR, users can plan, visualize, and control robot movements in a virtual space overlaid onto the physical environment. This approach enables greater precision in operations, reducing errors and improving the efficiency of collaborative tasks between humans and robots.

## 2.4 Conclusions

The analyzed literature provides a broad range of practical applications showcasing the potential of augmented reality (AR) in industrial contexts. The reviewed articles clearly demonstrate how AR can enhance human-machine interaction, optimize processes, and promote operational efficiency. However, these analyses are often limited to specific use cases, failing to provide a definitive solution to the broader challenge of integrating AR into industrial processes.

For each analyzed article, Table 1 summarizes the application field and implementation details, including the device used, the development environment, and the employed tracking method. This technological overview offers a valuable reference point for evaluating the tools available and most commonly utilized for this purpose. Notably, HoloLens is the predominant device used in nearly all cases, which justifies the selection of Unity as the game engine in this work due to its extensive support for different devices. Regarding tracking, Vuforia emerges as a preferred choice, and it will also be adopted in this work for its versatile tracking methods and ease of integration.

This overview highlights significant advancements in AR technology and its applications while also revealing critical gaps. Interoperability and standardization issues remain a key challenge for scaling AR solutions across diverse industrial ecosystems. Heterogeneous environments with legacy equipment, proprietary software, and varied data formats complicate seamless integration. This limitation underscores the pressing need for a systematic and interdisciplinary approach to address the barriers hindering widespread AR adoption.

In response to this need, the present work proposes the development of a dedicated framework. This choice is justified by the absence of structured guidelines supporting the real-time integration of data from mechatronic systems into AR applications. The framework aims to serve as a practical tool to address technological and organizational challenges, fostering a standardized and scalable approach to designing and implementing AR solutions in industrial settings.

Ref	Application	Device	Development	Tracking
[31]	Interaction	HoloLens	Unity	Fiducial markers/Vuforia
[40]	Monitoring	HoloLens	-	-
[41]	Monitoring	HoloLens-Smartphone	Vuforia Studio	QR Codes-Vuforia
[42]	Monitoring	Android devices	Unity	3D Model - Vuforia
[43]	Monitoring	HoloLens	Unity	Spatial anchoring
[44]	Monitoring	Hololens	Unity	Marker based -Vuforia
[45]	Monitoring	HoloLens	Unity	3D model - Vuforia
[46]	Monitoring	Android devices	Unity	Marker based - Vuforia
[47]	Monitoring	HoloLens/Ipad	Unity/Power BI	QR Codes
[48]	Monitoring	Apple device	Unity	3D Model-Wikitudo SDK
[49]	Training	AR Glasses	-	QR Codes
[50]	Maintenance	HoloLens	Unity	Marker based - Vuforia
[51]	Maintenance	Handheld devices	Unity	Plane det - ARCore
[52]	Maintenance	Android devices	Unity	QR Codes/Vuforia
[53]	Maintenance	Samsung Tab-Moverio BT-200	Unity	Marker based
[54]	Maintenance	Handheld devices	Vuforia Studio	3D Model -Vuforia
[55]	Maintenance	Smartphone	Unity	QR Code - Vuforia
[56]	Maintenance	HoloLens-Tablet Lenovo	Unity	Marker based - Vuforia
[57]	Maintenance	Android Smartphone	-	Marker based - Alvar Tracking SDK
[58]	Maintenance	Android devices	Unity	QR Code - VUforia
[59]	Maintenance	HoloLens	Unity	Image target- Vuforia

[60]	Maintenance	HoloLens	Unity	Image target - Vuforia
[61]	Maintenance	HoloLens	Unity	Proposed
[62]	Maintenance	Lenovo M10	PTC Creo	Marker based-Vuforia
[65]	Maintenance	Handheld devices	ARViewer	3D model
[67]	Maintenance	Android devices	Unity	3D model- Vuforia
[68]	Quality	HoloLens	Unity	3D model - Vuforia
[69]	IPT	HoloLens	Unity	-
[70]	Inspection	Android devices	Unity	Marker based - ARCore
[71]	HRC	HoloLens	Unity	-
[72]	HRC Assembly	Android device	Unity	Marker based /Vuforia
[73]	HRC	Android tablet	Unity - ROS	Marker based-Vuforia
[74]	HRC	HoloLens	Unity - ABB RS	Spatial anchoring
[75]	HRC	HoloLens	Unity	Marker based-Google's ARCore
[76]	Mobile Robots	Mobile devices	Unity - ROS	-
[77]	HRC	HoloLens	Unity - ROS	Vuforia

Table 1: Technological Summary of AR Implementations

## 3 Framework

### 3.1 Introduction and Motivation

In the previous Chapter [2], various contributions in the literature regarding the use of augmented reality (AR) tools in industrial applications were examined. Although current studies clearly define augmented reality as an enabling technology for a human-centered Industry 5.0 vision, several challenges have been highlighted that significantly limit its widespread adoption.

The identified issues have been divided into three main categories, as illustrated in Figure 1, and are grouped into factors related to the industry (the entity interested in adopting the technology), the technology itself, and the end user. Although much research focuses on improving technology and addressing the needs of the end user, the challenges related to integrating AR systems into industrial settings are not equally addressed. In this context, attention is often placed on practical examples of how AR integration can enhance processes, whether for monitoring, maintenance, assembly, or quality control, by developing custom systems to assess both their limitations and strengths.

This type of research undoubtedly raises awareness of the technology’s potential, but at the same time highlights one of its most critical challenges: the difficulty of integration. Each paper presents customised architectures for the specific application under discussion, which are highly dependent on the hardware and software components used and the purpose of the application.

Table 1 highlights the technologies used in the development of prototypes from recent articles discussing the use of augmented reality for industrial applications. Specifically, the articles focus on the device employed, whether it is an HMD (Head-Mounted Display) or HHD (Handheld Device), the software used for application development, and the tracking method implemented, whether marker-based or markerless. In the vast majority of cases, Unity is used for development due to its compatibility with almost all devices, while tracking is typically achieved with Vuforia, which offers an easily integrable package in Unity for image or 3D model tracking.

While these studies provide valuable insights into the development of AR systems, they often overlook critical aspects of integrating such systems into real-world industrial environments. Developing robust and scalable AR solutions for industrial settings presents unique challenges that go beyond what is discussed in the literature. The key among these challenges is the difficulty of adapting AR technologies to a wide range of existing machines, systems, and workflows without requiring costly or time-consuming modifications. Moreover, there is limited exploration of how AR solutions can be standardized or generalized for broader use cases.

Summarizing the industry-related factors highlighted in the literature review that limit the adoption of AR, the main issues identified are accuracy, reliability, cost and ROI, integration and interoperability, competence gap, and security. These factors are interrelated, and their resolution is crucial to enabling the effective adoption of AR technologies in industrial settings.

A deeper analysis of these limitations reveals that many of them stem from challenges associated with the integration of AR systems into existing industrial workflows and infrastructures. Addressing these integration challenges would not only directly improve factors like interoperability and scala-

bility but also indirectly enhance accuracy, reliability, and cost-effectiveness by reducing the need for extensive customization. Below, the key challenges that need to be tackled to support AR integration and address the broader industry-related factors are outlined:

- **Integration with existing machinery.** In all the reviewed cases, AR systems are developed in tandem with prototype machines, which are designed to work seamlessly with the AR application. However, for existing industrial machinery, there is little to no documentation or methodology provided on how AR can be retrofitted or integrated into current systems. This lack of guidance makes it difficult for industries to adopt AR solutions without extensive customization, which can be costly and complex.
- **Data representation.** A critical issue that is often overlooked is how machine data should be represented within the AR environment. In the reviewed literature, information of various types is simply linked to specific markers or 3D models. However, there is little discussion about how machines or systems with diverse data outputs can be represented generically, or how AR can handle complex data sets that are not easily associated with a single marker or object.
- **Definition of a generic architecture, independent of device and application.** Current AR applications are often highly customized to specific hardware and software setups, making it difficult to transfer solutions across different platforms or industrial contexts. There is a significant lack of research on how to create a generic, scalable AR architecture that could work independently of specific devices or applications. This lack of standardization limits the wider adoption of AR in industrial settings.
- **Establishing a clear development procedure for AR monitoring applications.** Another missing element in the literature is a well-defined process for developing AR-based monitoring applications. There is little guidance on the steps required to define specifications, choose appropriate hardware and software, and integrate these solutions into existing industrial processes. Without a clear roadmap, industries may find it difficult to navigate the technical and logistical complexities of adopting AR systems.
- **Scalability and interoperability challenges.** In addition to the points above, one of the main challenges in implementing AR in industrial contexts is ensuring scalability and interoperability between different devices, software systems, and industrial setups. Current AR solutions are often designed for specific use cases and lack the flexibility to adapt to different environments without significant rework.

This work seeks to tackle these challenges by establishing a comprehensive and innovative framework for the development of augmented reality (AR) interaction systems tailored to industrial environments. The framework is designed to provide a structured and scalable procedure that not only addresses the key challenges highlighted earlier but also establishes clear guidelines for implementing AR applications specifically tailored for monitoring and control tasks in mechatronic systems.

The AR application proposed in this framework serves two main purposes: **monitoring** and **control**. On the monitoring side, the application provides detailed information about all relevant system components, including their static characteristics (e.g., component descriptions and specifications) and dynamic attributes (e.g., real-time operational states). On the control side, the application enables users to interact directly with the system by exposing process-related control parameters. These include initiating specific procedures, configuring operational parameters, and managing workflows. This

dual functionality makes the application suitable for a wide range of industrial use cases, including **maintenance** (allowing operators to quickly access component information), **monitoring** (ensuring efficient oversight of system operations), and, more generally, enhancing **human-machine interaction** (enabling intuitive and responsive control mechanisms).

This work aims to address these challenges by defining a comprehensive and innovative framework for developing augmented reality (AR) monitoring systems in industrial contexts. The framework is designed to provide a structured and scalable procedure that not only addresses the key challenges highlighted earlier but also establishes clear guidelines for implementing AR applications specifically tailored for monitoring and control tasks in mechatronic systems.

The AR application proposed in this framework serves two main purposes: **monitoring** and **control**. On the monitoring side, the application provides detailed information about all relevant system components, including their static characteristics (e.g., component descriptions and specifications) and dynamic attributes (e.g., real-time operational states and performance metrics). On the control side, the application enables users to interact directly with the system by exposing process-related control parameters. These include initiating specific procedures, configuring operational parameters, and managing workflows. This dual functionality makes the application suitable for a wide range of industrial use cases, including **maintenance** (allowing operators to quickly access component information), **monitoring** (ensuring efficient oversight of system operations), and, more generally, enhancing **human-machine interaction** (enabling intuitive and responsive control mechanisms).

The framework is structured into three main points:

- **Defining the architecture and components:** The first step involves identifying the most suitable architecture for the AR monitoring system. Two primary approaches are analyzed: direct integration with the mechatronic system and the use of an intermediary server. Each option is evaluated for its advantages and drawbacks, such as simplicity versus scalability and adaptability. The key components of the architecture are also defined, detailing their functionalities and interactions. This ensures a clear understanding of what each element of the system must accomplish.
- **System design procedure:** After defining the architecture, the next step introduces a detailed procedure for defining the specifications of the system and its components. This procedure consists of three distinct phases:

1. **Parallel phases for AR and data definition:**

- The first phase focuses on defining the requirements and specifications for the AR application, including the necessary software and hardware components tailored to the specific use case.
- Simultaneously, an object-oriented model is developed to represent all relevant information about the mechatronic system. This model ensures consistency and adaptability, enabling the structured representation of data for any system.

These two processes can be executed independently, thanks to the intermediary server defined in the architecture, which manages the integration between the AR application and the mechatronic system, overcoming challenges such as interoperability.

2. **Final phase—Defining the server:** The last phase of the procedure focuses on defining the intermediary server. This server plays a critical role in facilitating communication between the AR application and the mechatronic system, addressing challenges such as data standardization, synchronization, and scalability.

- **Defining the conceptual model of the AR application:** The third and final point of the framework is the definition of a clear and generic structure for the AR application. This structure leverages the object-oriented model developed during the specification phase to dynamically generate and display information within the AR environment. By adopting the proposed framework, it becomes possible to create a reusable AR application suitable for any system that adheres to the defined information model. This approach ensures scalability, flexibility, and adaptability across different contexts.

The proposed framework offers several significant advantages:

- **Structured development procedure:** By providing a step-by-step process, it ensures clarity and consistency in developing AR applications for industrial use.
- **Adaptable data model:** The object-oriented information model organizes complex system data in a way that is adaptable to any mechatronic system, supporting scalability and standardization.
- **Generic AR application model:** A well-defined structure for AR applications ensures that they can dynamically generate information based on the underlying data model, enabling seamless adaptability across different industrial contexts.

In the following subsections, the framework will be defined, considering the following aspects. First, in 3.2, the most suitable architecture for the system will be identified, providing a comprehensive analysis of the advantages of using an intermediary between the augmented reality application and the mechatronic system to be monitored. Once the system components are established, the procedure for defining the specifications for the system’s implementation will be introduced and detailed in 3.3. This procedure highlights all the essential information required at each design phase, the outputs generated by these phases, and the dependency flow between the various steps. Finally, in 3.4, the conceptual model of the AR application will be presented.

## 3.2 Architecture definition

Defining the system architecture is a fundamental step in developing a robust framework for integrating augmented reality (AR) applications with mechatronic systems. The architecture establishes the foundation upon which all subsequent design and implementation decisions are based, ensuring a coherent and efficient approach to system development.

Before addressing the procedural aspects of implementation, it is essential to determine the structure of the system and the roles of its components. A well-defined architecture provides clarity on how the AR application interacts with the mechatronic system, manages data flow, and ensures compatibility between diverse technologies. Without this initial step, the design process risks being inconsistent or overly dependent on specific hardware or software configurations, limiting scalability and adaptability.

The architecture also directly influences the procedure by outlining the key functionalities and interfaces required. Furthermore, a clear architectural framework helps identify potential challenges early, allowing the design procedure to address them proactively, rather than reactively.

By defining the system architecture first, this work aims to establish a solid foundation that ensures the subsequent procedure is both comprehensive and aligned with the broader goals of creating a scalable and modular AR framework for industrial applications. This approach ensures that the procedure is not only systematic but also directly informed by the architectural choices, leading to a more robust and effective solution.

The system under examination, in its most basic form, is composed of two components: the mechatronic system of interest and the augmented reality-based applications designed to monitor and control the mechatronic system.

The information diagram of this interaction is shown in Fig. 2, highlighting the information exchange required in the application of interest between the two main components. On the left, the mechatronic system is depicted, conceptualized by considering the machinery (with sensors and actuators) as an entity that exchanges data with the Machine Controller, i.e., the computational and control unit. The controller interfaces with the system by reading data from the sensors and controlling the actuators. On the right, also in Fig. 2, the AR applications connected to the mechatronic system are shown, providing the user with visual information and allowing user input.

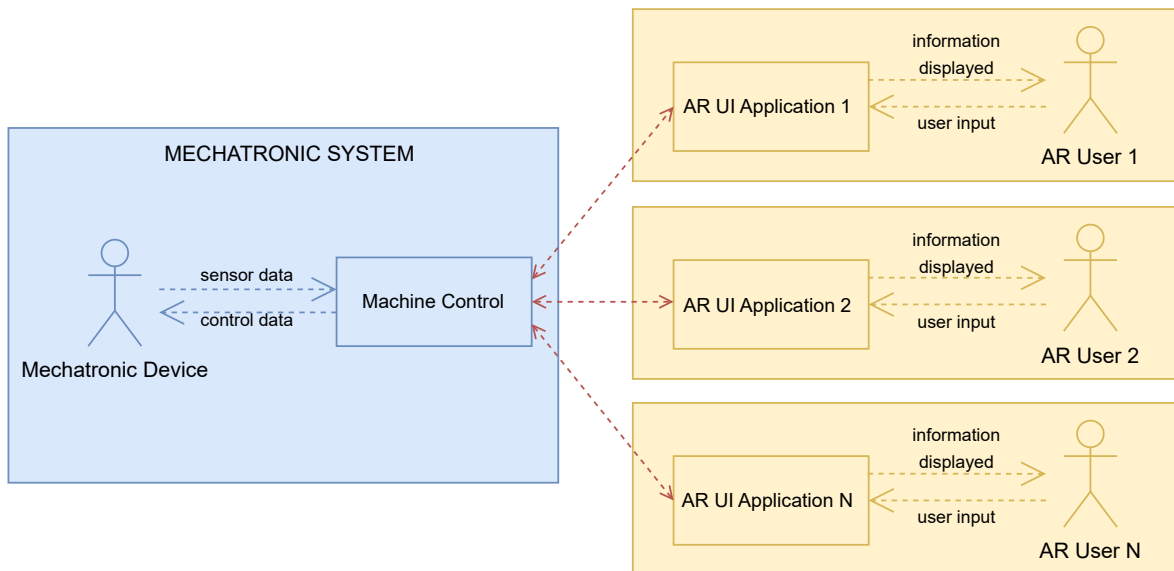


Figure 2: Information Diagram

The challenges highlighted in the literature regarding the creation of a direct-connection architecture, as represented in the figure, are numerous. Among them:

- **Compatibility:** Mechatronic systems and AR applications rely on different technologies and standards. The absence of an intermediary makes integration between systems that are not natively compatible more difficult. The direct connection might therefore be limited by the hardware capabilities of individual devices.

- Processing and computational load: In a serverless system, all data processing must be handled directly by the mechatronic system and the AR application. This can overload devices, especially if they need to process complex data in real-time or if their hardware resources are limited.
- Security: Without a centralized server, the responsibility for managing security falls entirely on the mechatronic system and the AR applications—this is a highly significant issue in industrial applications where security is paramount.
- Data Synchronization and Consistency: Without an intermediary to manage the global state, it becomes harder to ensure that the data displayed in the AR app remain consistent and up-to-date at all times.
- Limited scalability: Without a server, it becomes more challenging to scale the system to support multiple mechatronic devices or AR applications simultaneously. Each new device would require additional direct connections, potentially leading to management overhead.

Taking into account the points outlined, the inclusion of an intermediary between the mechatronic system and AR-based applications offers numerous advantages, making it the most suitable solution for the system architecture under development. Building on the premises introduced at the beginning of this chapter, the aim is to create a generic framework adaptable to various scenarios. This requires an architecture designed to ensure flexibility and modularity.

Figure 3 illustrates the architecture within the chosen configuration.

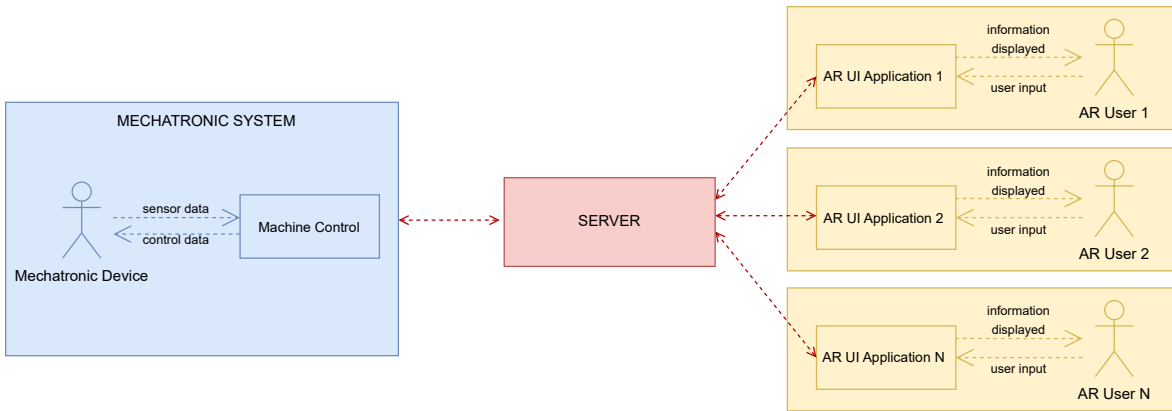


Figure 3: Information Diagram using an intermediary

Having defined the base architecture upon which the procedure will be developed, Figure 4 presents an exploded view of the subsystems, highlighting their main components. The representation follows the UML specification for structural diagrams. These diagrams illustrate a system's organization by depicting its components, such as classes, objects, and packages. They also represent the elements that constitute the system and their relationships. Specifically, the diagram used here focuses on showcasing the modular structure of the system and the interactions between its various parts.

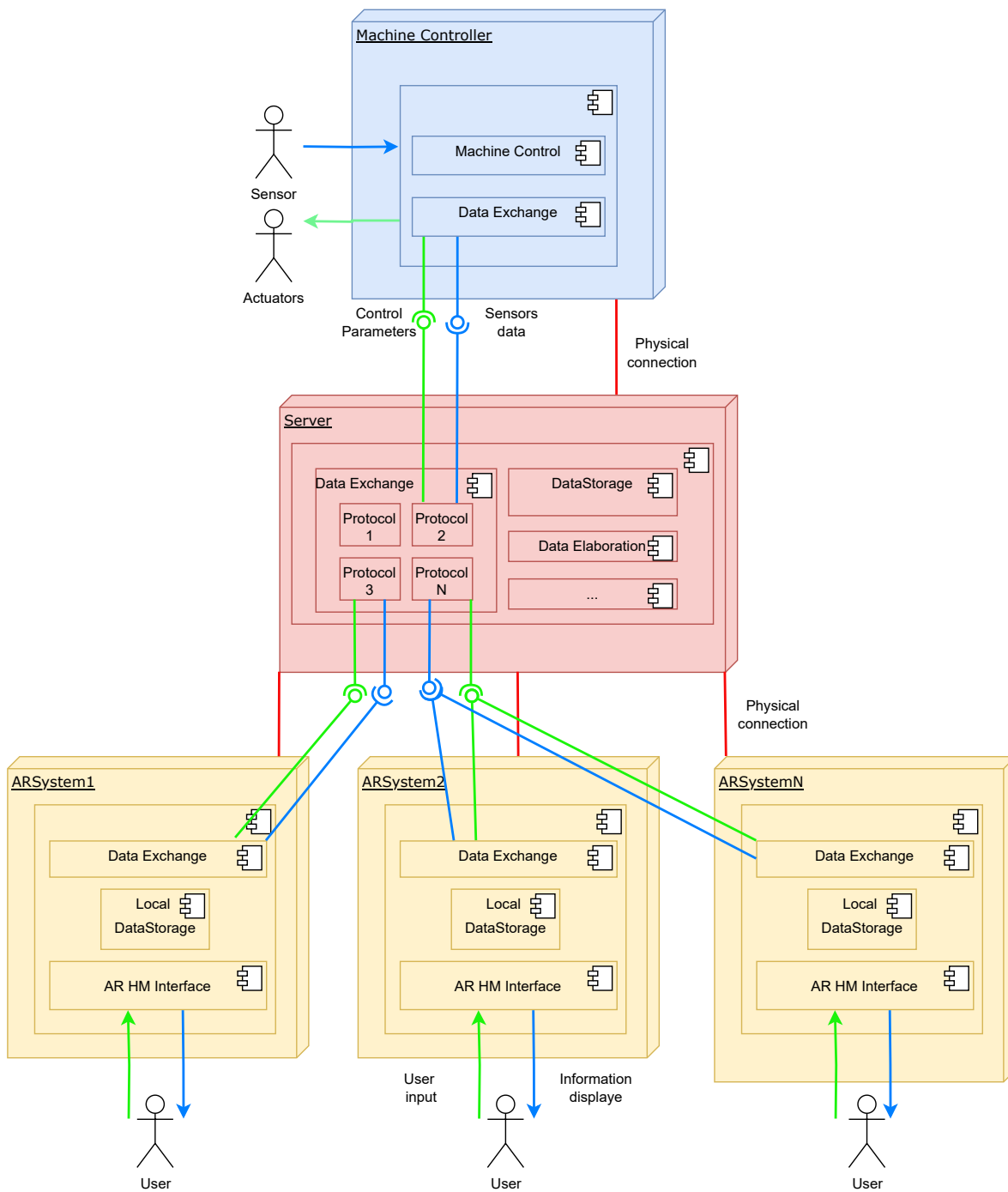


Figure 4: Structural diagram of the architecture, with subsystems and main components

The diagram includes three subsystems:

- The mechatronic system: made up of the machine controller and two key entities, which represent the components responsible for generating and consuming the data produced by the controller; these are the sensors and actuators.
- Augmented Reality applications, in the plural form, to emphasize the potential of this solution to enable the development of multiple applications connected to the same system.
- The central server, which acts as an intermediary between the two parts.

Each subsystem highlights its main components, as well as the interfaces it offers and requires to interact with other entities. In this context, each subsystem represents not only a software component designed to perform specific operations but also a division of the hardware infrastructure. Each subsystem is implemented on a separate device. For instance, the software components of the augmented reality systems will be developed for mobile devices in the case of handheld applications or head-mounted displays for HMD-based applications. Another example is the mechatronic system, which will feature dedicated hardware to control the machine, which will be extended via software to expose and request the data needed for the functionality required by the application.

With this premise, a dedicated subsection will provide a detailed description of each of the three subsystems.

### 3.2.1 The Mechatronic System

A mechatronic system can be defined as an integrated combination of mechanical, electronic, and computational components working together to achieve specific functions. It is the result of combining physical mechanisms with sensors, actuators, and control systems to create intelligent machines. These systems are built around a fundamental feedback loop: sensors monitor the system's behavior and provide data, the control system processes this information and determines appropriate actions, and actuators execute these actions to produce the desired output. This continuous interaction closes the so-called control loop, ensuring the system's performance meets the intended goals.

Building on this premise, the mechatronic system can be represented using an UML diagram including three main components:

- Sensors, which serve as the system's perception mechanism, monitoring its behavior and providing essential measurements such as position, speed, force, or temperature.
- Actuators, which are the system's action components, translating control commands into physical actions like motion or force generation.
- The Controller, which is the decision-making center that processes sensor data and sends commands to the actuators to achieve the desired objectives.

In the system diagram, both sensors and actuators are represented as actors following the UML notation. This representation highlights their active role in interacting with the controller and emphasizes their functional independence as entities that provide data (sensors) or execute commands (actuators).

The computational part is implemented as a subsystem called Machine controller, that represents a dedicated hardware device such as an industrial computer or a Programmable Logic Controller (PLC). The machine controller is composed by some software components, and always includes the Machine Control Module as main component, which is responsible for:

- Executing core control logic.
- Managing control loops for positioning the machine's axes.
- Ensuring safety mechanisms are in place.

The Machine Control Module represents the essential functionality found in most standalone mechatronic systems. However, when there is a need to connect the system to external entities, such as a central server, an additional component, must be integrated. This module is specifically designed to:

- Expose relevant operational data to external systems.
- Handle incoming data requests or commands from external sources.

In the diagram, this component is called: Data Exchange Module. This module is represented separately from the main component, the Machine Control Module, to emphasize that these functionalities can be added to extend the capabilities of an already existing machine with its own well-defined control software. It is unrealistic to assume that when augmented reality applications are developed to monitor data produced by a mechatronic system, the entire control logic of the system would need to be rewritten.

As discussed in the chapter on the literature review 2, one of the main challenges limiting the widespread adoption of augmented reality technology is precisely the difficulty of integrating it with pre-existing systems. By designing the Data Exchange Module as an independent and modular component, it becomes possible to interface new AR applications with legacy systems without disrupting their established control processes. This modularity ensures that the machine's control logic remains intact while enabling external systems, such as AR applications or central servers, to access relevant data seamlessly.

This module provides two logical interfaces to facilitate interaction with external systems:

- Data Provision Interface: This interface exposes the relevant data, which may include raw measurements from the sensors as well as processed data derived from the control software's logic.
- Control Parameter Interface: This interface allows external systems to input parameters to modify the control logic of the system.

At first glance, the second interface might seem closely tied to the system's control logic and not entirely separate from it. However, the underlying idea builds on the analogy of a graphical interface typically used by operators to input commands. In this case, such commands are provided externally, by connecting the incoming data from the Control Parameter Interface to the internal variables linked to the graphical interface. This approach enables external systems to adjust control parameters without disrupting the core control logic.

Defining logical interfaces means specifying how data is requested and made available, including the format, structure, and expected behavior of the data exchanges. This clear definition ensures that, if needed, the Machine Control Module can be modified to utilize data provided by these interfaces as parameters. Since the type and format of the published data are well-defined, integration becomes more predictable and manageable.

By designing such interfaces, the system achieves a balance between modularity and adaptability, making it possible to integrate external functionalities, such as augmented reality applications or centralized monitoring systems, without requiring significant modifications to the existing control logic. This modular and structured approach supports interoperability while maintaining the integrity and reliability of the mechatronic system.

### 3.2.2 Augmented Reality Applications

Augmented Reality (AR) applications can serve various purposes in the industrial context. As highlighted in the state-of-the-art analysis (2), there are numerous examples of AR-based applications designed to assist operators in tasks such as quality control, maintenance, assembly, or, more broadly, the development of human-machine interfaces that facilitate more effective and intuitive data exchange. Despite the wide range of potential applications, common components exist across all AR implementations, simplifying their conceptualization and design.

In this architecture, the representation uses a UML diagram to depict a subsystem composed of the primary software components, connected to an actor representing the end user. The ARSystem subsystem, shown in the diagram as a repeated instance to emphasize the possibility of a multi-slave configuration within the proposed architecture, corresponds to the hardware on which the AR application is deployed. This hardware could be a handheld device or a head-mounted display (HMD).

The subsystem is structured around three main components:

- **The Data Exchange Module:** Similar to the module in the mechatronic system, this component provides interfaces to publish and receive the data required by the application. Its role is to enable seamless communication between the AR system and the central server or other subsystems.
- **Local Data Storage:** This component stores static information about the mechatronic system that does not change over time. For instance, in monitoring applications, the user might need to access properties of a specific device within the system. These properties are static and do not require cyclical communication between the server and the application. Providing this information during the setup phase allows the application to store it locally in a dedicated database for on-demand use, ensuring more efficient system performance.
- **The AR Human-Machine Interface (AR HMI):** This module is responsible for creating the user interaction interface. Represented in the diagram with a direct connection to the actor symbolizing the end user, this module encompasses all activities related to the consumption and production of data presented to the user. It addresses how data is displayed, how users interact with it, and, critically, how information is spatially positioned in the augmented environment.

The last component dealing with the representation of information in augmented reality therefore includes one of the foundational activities common to all AR applications, regardless of their specific purpose, the localization. Accurately determining and tracking the device's position relative to the machine is essential for developing effective AR applications. This can be achieved using various methods, either marker-based or markerless. Marker-based systems rely on predefined visual or physical markers to establish positional references, while markerless systems use techniques such as feature detection, simultaneous localization and mapping (SLAM), or external tracking systems. The choice of localization method depends on the application's requirements and the operational environment.

By structuring AR applications around these components, the architecture ensures modularity, scalability, and adaptability, enabling efficient integration into industrial workflows and seamless interaction with existing mechatronic systems.

### 3.2.3 The Server

In the introductory section of this chapter, the limitations of using a direct connection architecture between AR applications and the mechatronic system were analyzed in detail. Consequently, the diagram includes a server that acts as an intermediary between the two components described above.

This server is depicted as a subsystem, similar to the other subsystems, comprising independent hardware and the software modules it hosts.

Among these modules, the Data Exchange Module is pivotal in establishing a robust communication framework between the server and the various systems it interacts with. This module is designed to support multiple communication protocols, each tailored to the specific needs of the connected clients. Examples of such protocols might include HTTP/REST or WebSocket for general-purpose data exchange, OPC UA for industrial interoperability, or even proprietary protocols optimized for high-speed or low-latency operations.

By implementing a variety of protocols, the Data Exchange Module ensures compatibility with a wide range of client systems, including AR applications, mechatronic controllers, and other subsystems. These protocols expose well-defined interfaces, which are the access points for data exchange. Each interface specifies the format, structure, and semantics of the data, as well as the procedures for requesting, sending, or receiving it.

This architectural approach emphasizes the server's role to ensure:

- **Modularity:** New protocols or interfaces can be added to the server without disrupting its existing functionalities, enabling seamless adaptation to evolving requirements.
- **Flexibility:** By supporting diverse communication standards, the server can interact with heterogeneous systems, making it suitable for integration into complex industrial environments where multiple technologies coexist.
- **Extensibility:** The ability to expose additional interfaces or protocols ensures that the architecture remains scalable, allowing for the inclusion of new clients or functionalities without significant reengineering.

In practice, the Data Exchange Module acts as a bridge between systems that might otherwise be incompatible. This multi-protocol capability not only enhances interoperability but also future-proofs the system, enabling it to adapt to new technological trends and integrate with emerging industrial standards. As such, the Data Exchange Module is not just a data exchange channel but a foundation of the scalability and resilience of the architecture.

A second key module is the Data Storage Module, which plays a crucial role in bridging the gap between the data-generating system and the AR application. This module serves as a centralized repository for both static data and dynamic data, each serving distinct purposes within the architecture.

Static data refers to information that does not change over time or changes only infrequently. Examples include:

- **Properties of devices:** Specifications such as model numbers, capabilities, calibration settings, or operational limits.

- System configurations: Predefined settings that govern how the system operates, such as network parameters or control logic configurations.

Storing this information centrally on the server ensures it is easily accessible to all connected clients, including AR applications. This approach enhances efficiency by eliminating the need to repeatedly transmit static data, thus reducing network bandwidth usage. Instead, static data can be provided to the AR application during the setup phase and cached locally for on-demand use. It also ensures consistency, as a central repository guarantees that all clients access the same version of the static data, avoiding discrepancies caused by outdated or inconsistent information across devices. Furthermore, centralizing static data storage supports scalability, allowing new AR applications or other clients to retrieve the necessary information directly from the server without placing additional queries on the mechatronic controller or other subsystems.

Dynamic data encompasses information that changes in real-time or near-real-time, such as: sensor readings and control signals. The Data Storage Module serves as an intermediary for dynamic data by managing the flow of information between the mechatronic system and AR applications. Rather than having AR applications poll the mechatronic system directly, the server handles data acquisition and makes it available to clients, reducing the computational load on the mechatronic controller and streamlining communication.

The final component shown is the Data Elaboration Module, which processes raw data from sensors to generate meaningful insights, such as filtering noise or calculating derived metrics. The significance of this module lies in its ability to centralize computational demands, thereby reducing the processing load required by client systems. This centralized approach enhances the efficiency and scalability of the architecture while ensuring that the computational power of AR devices or other clients can be reserved for tasks directly related to user interaction or visualization.

By integrating these modules, the server not only facilitates seamless communication and data exchange between components but also provides additional functionalities that strengthen the overall system's robustness and adaptability to future requirements.

### **3.3 Design Process**

In the previous subsection, the architecture for developing a generic system leveraging augmented reality (AR) for monitoring and controlling a mechatronic system was defined and justified. It was demonstrated that incorporating a server as an intermediary between the machine and the AR application provides significant advantages in terms of reliability, flexibility, and scalability.

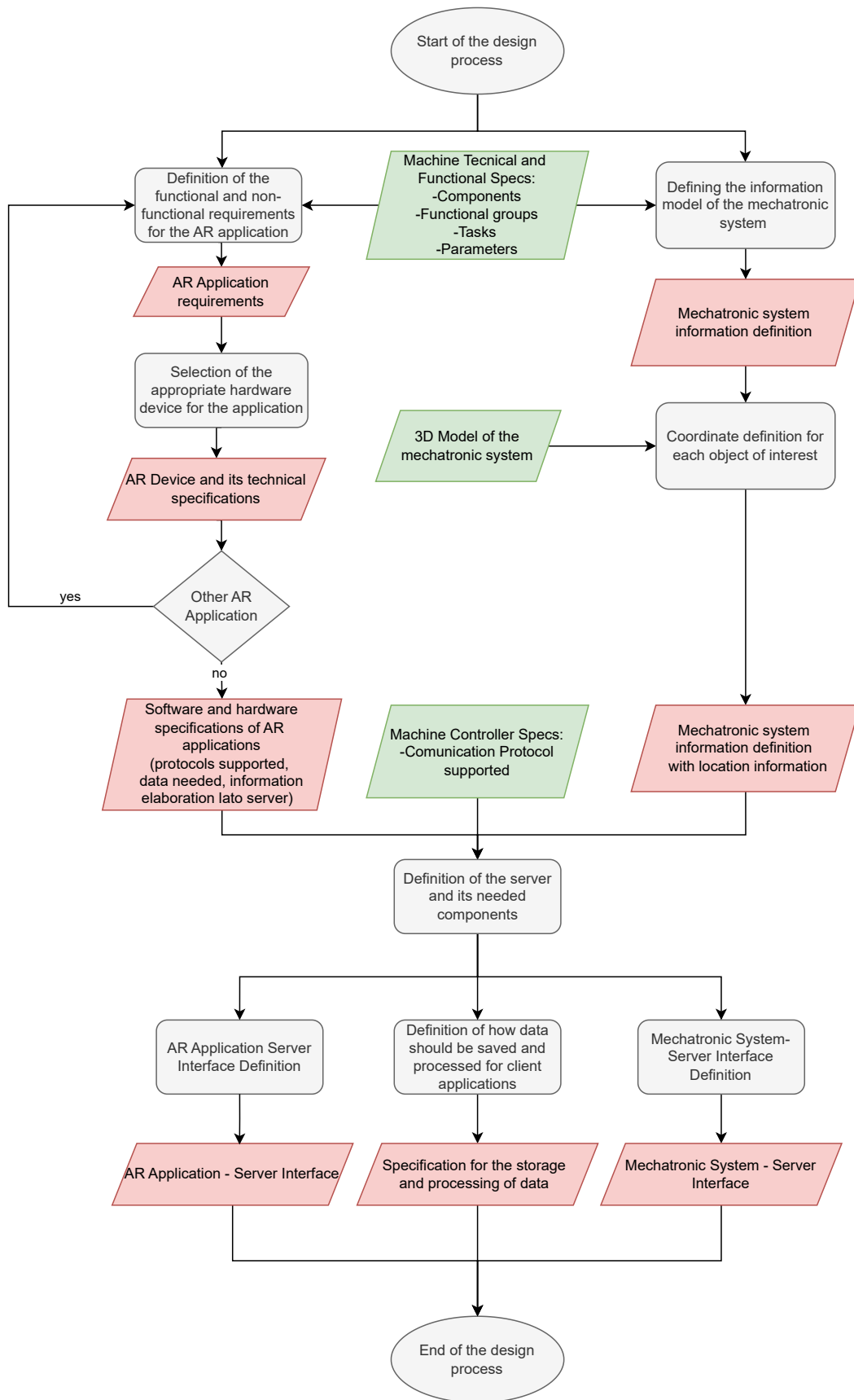


Figure 5: Flow chart of the design process

Alongside the representation of the system's main components 4 (the mechatronic system, the server, and the AR applications), their primary sub-components were also detailed, highlighting the functionalities necessary to fulfill their roles.

The developed architecture was designed to address most challenges associated with such systems, as discussed in depth in the respective subsections. While the architecture lays the foundational framework for building these systems, it alone is insufficient to overcome the well-documented difficulties in implementing them in industrial contexts. The design choices that need to be made based on this architecture are still complex and demand considerable time and expertise, making them incompatible with typical industrial constraints.

Once the architecture has been established-defining its hardware and software components and the connections between them-it becomes necessary to outline a procedure for moving from the conceptual framework to its practical definition. This procedure provides clear guidelines for bridging the gap between the conceptualized architecture and its implementation.

The proposed procedure details the steps required to identify the specifications for system implementation, emphasizing the critical information needed at each stage and the deliverables produced. It also offers a sequential definition of the problem, accounting for the inter-dependencies and mutual influences of the decisions made throughout the process. This structured approach ensures a well-defined path from design to deployment, enabling the development of a robust and efficient AR-enabled mechatronic system that can meet the demands of industrial applications.

The outcome is depicted in Figure 5. The diagram consists of blocks of various types, each with distinct meanings. Oval blocks represent the start and end states of the design process. Rounded rectangles indicate the various procedural stages, each representing independent decision-making phases that require specific input data and produce partial outputs. Green parallelograms represent the input data essential for the process. These inputs include the technical and functional specifications of the mechatronic system to be monitored, along with the specifications of its controller and its 3D model for spatial information. Conversely, red parallelograms represent information generated by intermediate stages of the procedure. These intermediate outputs are subsequently consumed by later stages to produce the final outputs of the process.

As illustrated in the diagram, the procedure's flow can be conceptually divided into three parts. The first two parts can occur concurrently: one focuses on defining the AR applications, while the other builds the information model of the mechatronic system. The outputs of these two concurrent stages provide all the necessary information for the final step, which involves defining the specifications of the server that acts as an intermediary within the system. As previously introduced, using an intermediary simplifies the integration of two systems from different domains, enabling the earlier stages to be decoupled and developed independently. This highlights why the server specifications must be determined after making decisions about the two primary components to be connected.

A notable aspect of this framework is its approach to information representation. Unlike the methods commonly described in the case studies reviewed in the literature review (Section 2), this framework proposes a generic, object-oriented model for representing a mechatronic system. The goal

is to ensure that the information is not merely linked to markers in augmented space but is structured in a manner that enhances its usability and relevance for the operator. The definition of a model represents also a key point in the development of the application, which can thus be compatible with any system as long as the information is modelled using the proposed information model.

In the following subsections, the procedure will be explained in detail, breaking it down into three key phases: defining AR applications, modeling the information of the mechatronic system, and specifying the server.

### 3.3.1 AR Application design phase

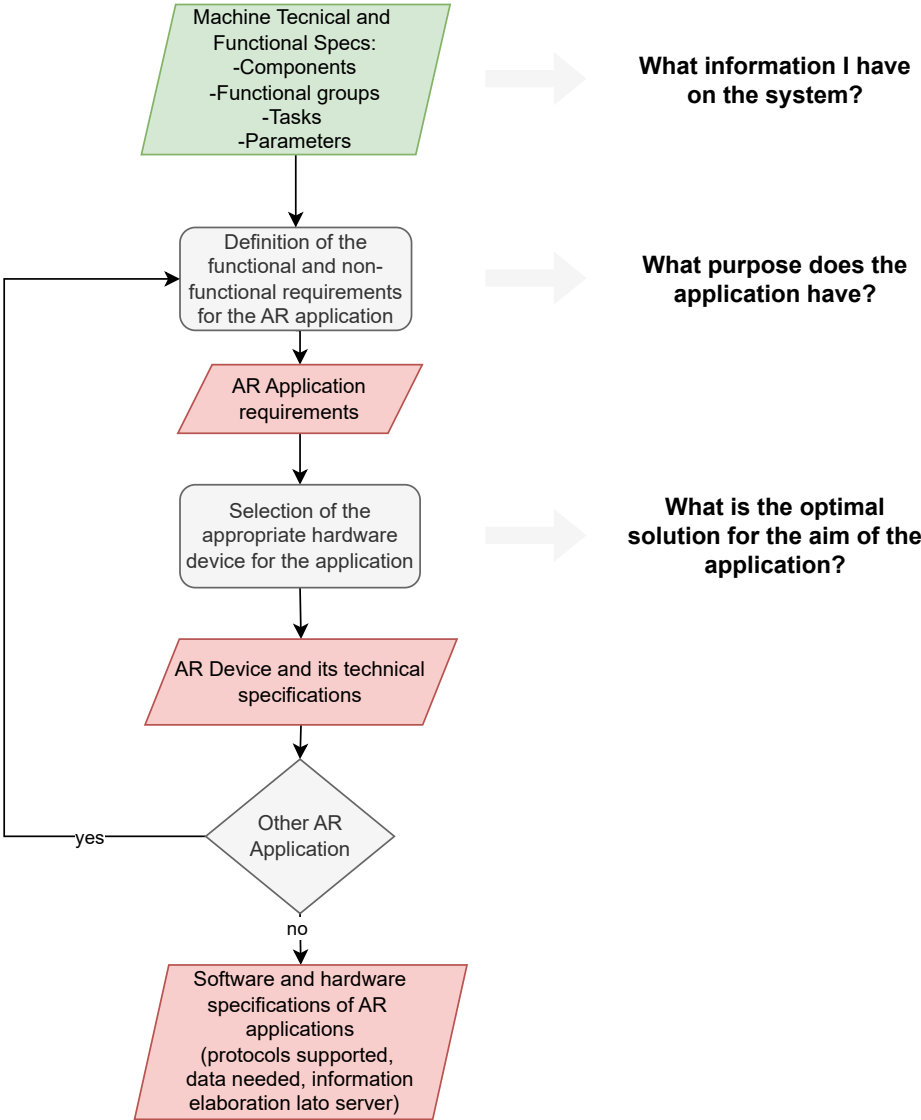


Figure 6: AR Application design phase

In the chapter dedicated to the state of the art [2], the use of Augmented Reality (AR) in industrial environments has been identified as an enabling technology for improving the efficiency of various operations. Being able to visualize machine information in strategic locations can enhance fault detection, simplify plant maintenance, and generally improve the monitoring process for system operators.

When designing an AR application, some key aspects must be addressed to determine the best implementation strategy. Figure 6 illustrates the part of the workflow diagram related to defining an AR application. As previously introduced, this phase is entirely independent of others due to the intermediary role of a server.

The green parallelogram represents the information known and necessary for the procedure, i.e. the technical and functional specifications of the machine with which the application will interact. These are the only inputs and represent the context for which the AR application is being developed.

The first step in the procedure (the initial rounded rectangle in Figure 6) involves defining the objectives of the AR application, generally expressed as functional and non-functional requirements. Functional requirements ensure the system performs the necessary tasks, while non-functional requirements guarantee reliability, security, and efficiency. The output of this phase (red parallelogram) includes a detailed set of application requirements.

Table 2 summarizes some key aspects to consider when defining these requirements. Notably, these are presented in a way that is easily understandable for stakeholders, without delving into the implementation specifics of the application.

<b>Aspect</b>	<b>Requirements</b>
<b>Use Context</b>	<ul style="list-style-type: none"> <li>- Purpose of the application (monitoring and control)</li> <li>- Indoor or outdoor environment</li> <li>- Duration of use per session</li> </ul>
<b>Key Features</b>	<ul style="list-style-type: none"> <li>- Type of AR (marker-based, markerless, or spatial mapping)</li> <li>- Tracking precision requirements</li> <li>- Visualization of 3D models</li> </ul>
<b>User Interaction</b>	<ul style="list-style-type: none"> <li>- Interaction methods (e.g., controller, hand gestures, voice)</li> <li>- Required input/output mechanisms (e.g., tactile feedback, audio)</li> </ul>
<b>Integration</b>	<ul style="list-style-type: none"> <li>- Compatibility with external sensors or industrial systems</li> <li>- Communication with middleware or server</li> </ul>
<b>User Ergonomics</b>	<ul style="list-style-type: none"> <li>- Comfort for prolonged use</li> <li>- Adaptability for various user profiles</li> </ul>
<b>Robustness</b>	<ul style="list-style-type: none"> <li>- Resistance to industrial conditions (e.g., dust, vibrations, temperature)</li> <li>- Stability of tracking in dynamic environments</li> </ul>
<b>Target Audience</b>	<ul style="list-style-type: none"> <li>- Expertise level of the end-user</li> <li>- Simplicity of the user interface</li> </ul>

Table 2: Functional and Non-Functional Requirements for Industrial AR Applications

The ‘Context of Use’ section emphasises the definition of the aim, in this case generically defined by the framework, i.e. monitoring and control. Once the general purpose has been defined, additional and specific operating requirements are defined through the definition of the other requirements. Application details may be specified, for example the environment, the duration of use, the tracking method to be used and the interaction methods available to the user.

Requirements must then be translated into the technical specifications necessary to achieve them. This step is crucial for defining the constraints the hardware and software must satisfy. The 3 table

provides a correlation between the functional and non-functional requirements of the 2 table and the technical specifications that could be derived from them.

Table 3: Translation of Functional Requirements into Technical Specifications

<b>Requirement</b>	<b>Technical Specification</b>
<b>Use Context</b>	
Purpose of the application (e.g., monitoring, control)	High-performance AR headset capable of real-time updates. Head-mounted or handheld devices
Indoor or outdoor environment	IP rating for dust and water resistance; brightness adaptable to industrial lighting.
Duration of use per session	Battery life supporting at least 4 hours of continuous use.
<b>Key Features</b>	
Type of AR (e.g., marker-based, markerless, spatial mapping)	Support for Vuforia or equivalent SDK; depth camera for markerless AR and spatial mapping.
Tracking precision requirements	Inside-out tracking with SLAM; high-resolution depth sensors.
Visualization of 3D models	GPU supporting smooth rendering of complex 3D models.
<b>User Interaction</b>	
Interaction methods	Support for hand gesture recognition (e.g., MRTK); voice recognition system; compatibility with external controllers.
Required input/output mechanisms	Haptic feedback module; integrated microphone and speakers for audio I/O.
<b>Integration</b>	
Compatibility with external sensors or industrial systems	Support for OPC UA or MQTT; USB or Bluetooth for external sensor integration.
Communication with middleware or server	Wi-Fi 6 or industrial-grade Ethernet for seamless connectivity.
<b>User Ergonomics</b>	
Comfort for prolonged use	Lightweight head-mounted display (<500g); adjustable head straps for various user profiles.
<b>Robustness</b>	
Resistance to industrial conditions	Shockproof, dustproof, and heat-resistant materials.
Stability of tracking in dynamic environments	Advanced motion compensation algorithms; accelerometer and gyroscope integration for stability.
<b>Target Audience</b>	
Expertise level of the end-user	Intuitive user interface with simplified navigation. Head-mounted or handheld devices.

<b>Requirement</b>	<b>Technical Specification</b>
Simplicity of the user interface	On boarding tutorials integrated into the AR application.

Many specifications directly relate to hardware choices. For instance, IP ratings for industrial environments, lightweight head-mounted displays, and high-performance batteries are physical attributes. Other specifications address software needs, such as the integration of Vuforia for marker-based AR or advanced motion compensation algorithms for dynamic environments. Tracking precision and visualization capabilities (e.g., GPU power) lie at the intersection of hardware and software, requiring careful evaluation of both. The separation between user-level requirements and technical details allows for an iterative refinement process, ensuring alignment between application goals and feasible technological solutions.

Defining both requirements and technical specifications enables a clear understanding of the hardware needed for the AR application. This phase results in a structured list of technical specifications, which can be compared against the features of available devices on the market. This not only facilitates informed purchase decisions but also provides a framework for identifying gaps where custom solutions may be necessary.

By systematically following these steps, it becomes possible to establish clear guidelines for designing AR applications aimed at monitoring and controlling mechatronic systems. The structured approach helps stakeholders evaluate trade-offs between various options, such as selecting between head-mounted or handheld devices, prioritizing robustness over lightweight design, or choosing software frameworks based on tracking precision requirements. Ultimately, this methodology ensures that the final application aligns with both user needs and operational constraints, streamlining its adoption in industrial contexts.

### 3.3.2 Mechatronic system information modelling phase

As outlined in the overall procedural framework, the process can be divided into three main phases: two concurrent phases related to system modeling (discussed in this section) and the definition of AR specifications, followed by the final phase concerning the server. In parallel with defining the requirements and technical specifications for AR applications, it is equally crucial to determine how the information about the mechatronic system should be structured and represented.

An analysis of the state of the art reveals that typical solutions do not focus significantly on the information structure; instead, the data is often linked to physical locations of interest without providing a systematic or structured definition.

In the proposed procedure outlined in this work, a well-defined model is introduced to represent a generic mechatronic system. The aim is to organize the system's information effectively and efficiently to serve the intended purpose. The definition of this information scheme will be the basis for defining a generic AR monitoring and control application, based on the model with which the information is encapsulated, becoming system-independent and reusable.

Figure 7 highlights the section of the procedure that illustrates this phase.

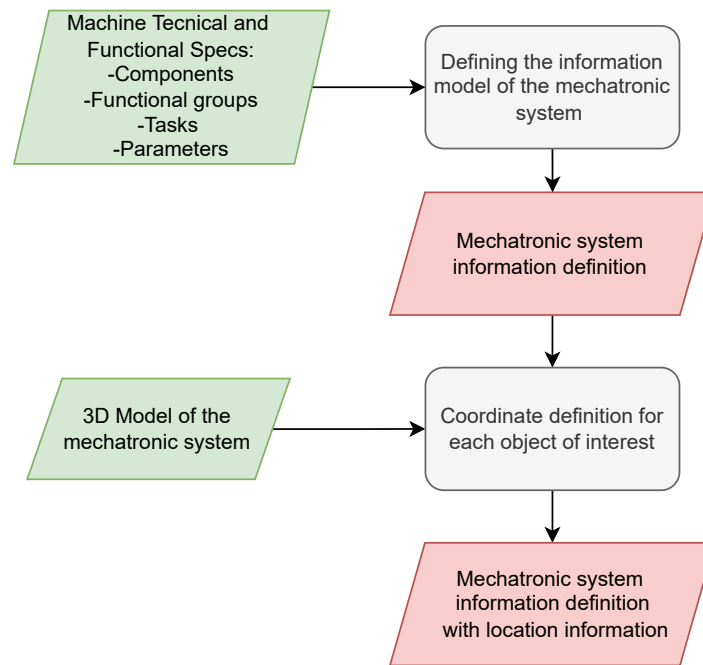


Figure 7: Mechatronic system information modeling phase

The starting point of this procedure is the set of available information about the mechatronic system, which includes details about its components, their functions, the data generated and consumed, and, more broadly, all the information expected to be available to those developing an AR-based monitoring system. Mechatronic systems inherently comprise a vast array of diverse information, and defining a generic schema to structure this information would facilitate its effective utilization.

From this information, the objective is to establish a model of the mechatronic system and its data (represented by the first rounded rectangle in the figure). The output of this step is referred to as the Mechatronic System Information Definition. Considering the purpose of the procedure—enabling the system for integration with AR applications—the next step involves identifying key positions on the system where information should be visualized. To achieve this, the information model must be enriched with the spatial data of the various system components.

There are multiple methods to gather spatial data. While measurements can be taken directly from the physical machine, the most effective approach involves leveraging a 3D model of the system. It is assumed that such a model is always available in the development context.

The final output of this section of the procedure is a schema that encompasses all relevant information, augmented with the spatial coordinates of the various components.

This section provided a generic overview of the steps to be followed. The next part will delve into the specific model proposed for data representation, offering a detailed explanation of its structure and implementation.

A mechatronic system can be viewed as a collection of components that collaborate and coexist to fulfill specific functions. Adopting the object-oriented modeling paradigm, a single component can be considered a fundamental building block that, when connected to others, forms the entire system.

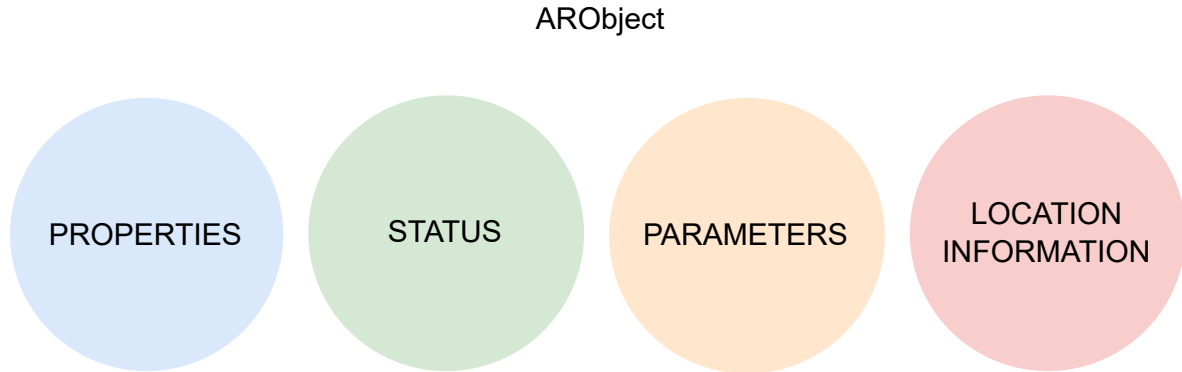


Figure 8: ARObject main components

This fundamental object, termed ARObject, can be defined as a set of four categories of data, as illustrated in Figure 8. Specifically, a generic component includes:

- **Properties:** Static information that characterizes the component and does not change over time. Most components will have standard properties, such as the manufacturer, product code, and identifier, alongside custom properties specific to the component.
- **State:** Dynamic information that changes over time and defines the component's condition at a given moment.
- **Parameters:** Non-static attributes that users can modify or configure.
- **Position Information:** Spatial data necessary for locating the object in an augmented scenario.

A class diagram representing these concepts is shown in Figure 9. Here, the ARObject encapsulates the four main categories described above. According to UML standards, relationships in the diagram are defined as compositions, highlighting that the sub-elements exist only as part of the object containing them.

The diagram specifies the cardinality of these relationships:

- **Position Information:** Mandatory (cardinality = 1), as every ARObject intended for augmented reality visualization must have a defined position.
- **Properties:** At least one property (cardinality = 1-N) must be defined: the object's identifier.
- **State and Parameters:** Optional (cardinality = 0-N), as some components might not have parameters to configure or a state to monitor.

The diagram also introduces additional classifications. Properties are divided into two subcategories: intrinsic properties and functional properties. Intrinsic properties represent characteristics inherent to the component, such as the object's identifier, manufacturer, or product code. Functional

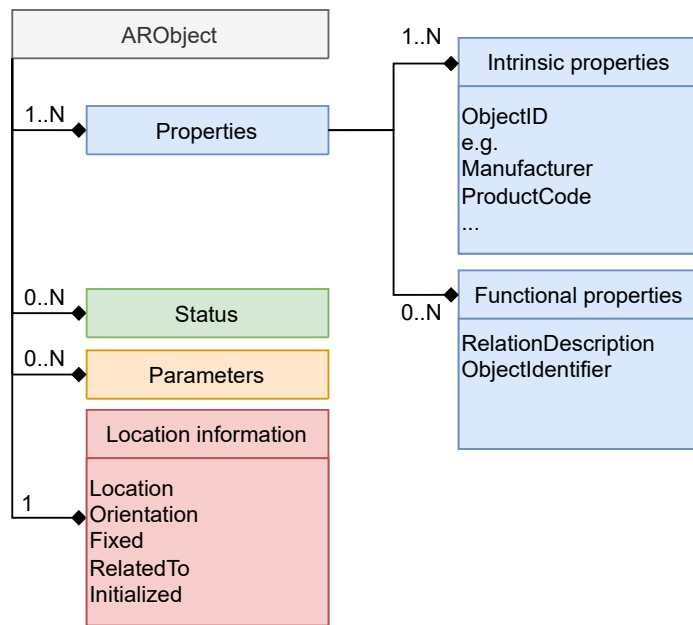


Figure 9: UML class diagram of ARObjct

properties describe functional relationships between objects. For example, they include the relationship between a sensor and the object whose property it measures or, in a multi-axis system, the relationship between a motor and the axis it moves. These properties provide a way to establish connections between objects, offering users insights into the functional interdependencies within the monitored system.

The localization information is structured to include coordinates  $(x, y, z)$  that represent the object's position in space, its orientation, and a flag indicating whether the object is fixed or dynamically changes its position based on the system's state. For instance, a sensor mounted on a moving axis will modify its position with the axis's movement. Additionally, the `RelatedTo` attribute identifies the reference coordinate system. This facilitates coordinate transformations, which are especially useful when leveraging the known kinematics of a system to compute an object's position. The `Initialized` attribute specifies whether the object's position has been determined. This is particularly relevant for components whose positions vary and are determined by a non-absolute encoder. In such cases, initialization occurs during the sensor's zeroing phase.

Having defined the main element, the `ARObjct`, the mechatronic system could be represented as a list of objects with their associated information defined as described. However, considering the complexity of such systems, this approach would limit the representation to individual objects. While functional properties are defined to establish connections, this is not sufficient to provide a comprehensive description of the system.

To create a generic information model, adaptable to different systems and capable of offering applications a more exhaustive representation of the complexity of a mechatronic system, a model is proposed that integrates `ARObjcts` into a broader descriptive structure, as represented in Figure 10.

The information model described provides a modular and functional representation of a mechatronic

system, organized into functional groups that represent subsystems responsible for specific functions of interest. Each functional groups may include sensors, actuators, mechanisms, and other physical or logical entities, forming an autonomous but interconnected part of the overall system. This structure enables the system to be segmented into units with clearly defined functions, facilitating both the understanding of internal relationships and the diagnosis of issues in the event of malfunctions. Also in this overall schema, the cardinality of the aggregation of components is specified. In most cases, it is defined as 0..N, indicating that the object is not mandatory.

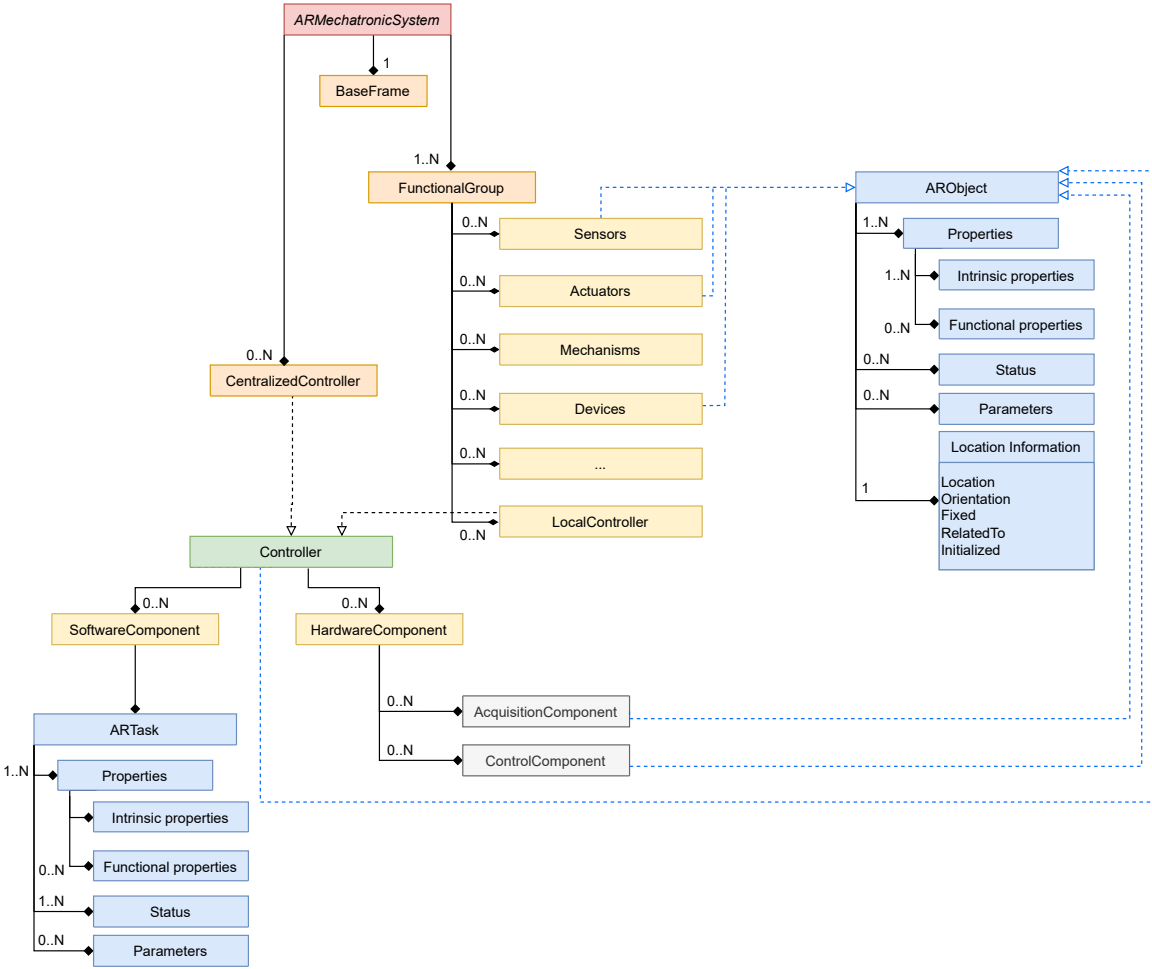


Figure 10: Comprehensive information model

ARObjects form the foundational building blocks of the model and are used as a base class to construct more specific classes representing sensors, actuators, and other hardware entities. Each ARObjct inherits a structure that includes intrinsic properties (e.g., identifiers and static characteristics), functional properties (defining relationships between components), a state (dynamic information), and configurable parameters. Additionally, every ARObjct must have localization information, including position, orientation, and attributes to manage spatial referencing, such as the origin of the coordinate system and an indication of whether the component is fixed or mobile.

The model supports both centralized and local controllers. Local controllers manage specific subsystems, reducing the complexity of centralized control and enhancing modularity. Each controller can

have hardware components, such as acquisition boards, and software components that represent tasks.

With the aim of enabling control of the system, the ARTask object is defined, which is conceptually associated with controllers, local or centralised, and is used to describe specific activities within the system. Each task has a set of properties, a state, and control parameters. The task's properties define its static characteristics, such as the type of activity or relationships with other system elements. The task's state includes the process variables being monitored, providing an up-to-date view of the activity's operational conditions. Parameters, on the other hand, are configurable values that control or modify the task's behavior. This approach not only allows for the monitoring of the mechatronic system but also offers an integrated view of the processes it supports, enabling dynamic management of operations.

The classification into functional blocks, combined with the possibility of defining functional properties for each component, makes it possible to specify which functional group performs a given task. This possibility of representation is crucial for establishing a direct link between tasks and the components responsible for their execution. In the event of problems with a task, this relationship allows the components of the functional group involved to be quickly monitored and analysed, supporting maintenance, diagnostics and optimisation activities.

This representation of the mechatronic system goes beyond modeling static data and relationships between components. It also provides a means to integrate the dynamic dimension of the process, ensuring detailed monitoring and control. The modularity of the model and the use of flexible base objects allow for the representation to scale seamlessly from simple to complex systems, maintaining a balance between technical detail and ease of management.

Once the information of the mechatronic system has been defined following this reference model, it will be implemented on the server, which will then contain all the system's static information and provide interfaces to update the dynamic information.

### 3.3.3 Server design phase

As previously mentioned, this procedure is divided into three main parts. The first of the two concurrent steps determine the specifications for the AR applications, including their hardware and software requirements, while the second part defines the information model for the mechatronic system.

The outcomes of the first two part of the procedure, combined with the known information about the mechatronic system—specifically the technical details of the machine controller with which the server will interface—are crucial in deciding the server's specifications. This ensures that the server's design aligns with both the AR applications' needs and the system's functional constraints.

In Figure 11, the relevant section of the procedure is shown.

Among the decisions that must be made to fully define the server and its components are:

- **AR Application-Server Interface:** This defines how the server will communicate with the AR application. Based on the specifications derived from the design procedure of the AR application, the requirements of the selected device are determined. Depending on the protocols supported by the device, an interface can be developed to enable communication between the two components.

- **Definition of Data Storage and Processing:** Once the information model of the mechatronic system has been defined, it must be implemented on the server. If the AR application requires specific data processing, the server should handle it to reduce the computational load on the AR device. This ensures efficient performance and offloads complex calculations to the server.
- **Mechatronic System-Server Interface:** Similarly, based on the protocols supported by the mechatronic system’s controller, the communication interface between the two systems must be defined. This ensures seamless interaction between the server and the mechatronic system for real-time data exchange and monitoring.

Once these considerations are addressed, each component of the architecture initially presented in Figure 4 is well-defined. At this point, the next step is the implementation of the design to finally develop the system.

To facilitate this, the framework proposes a structure for the development of the AR application. This structure ensures the creation of an application that complies with the proposed data model and is reusable. Further details on this framework are presented in the next section.

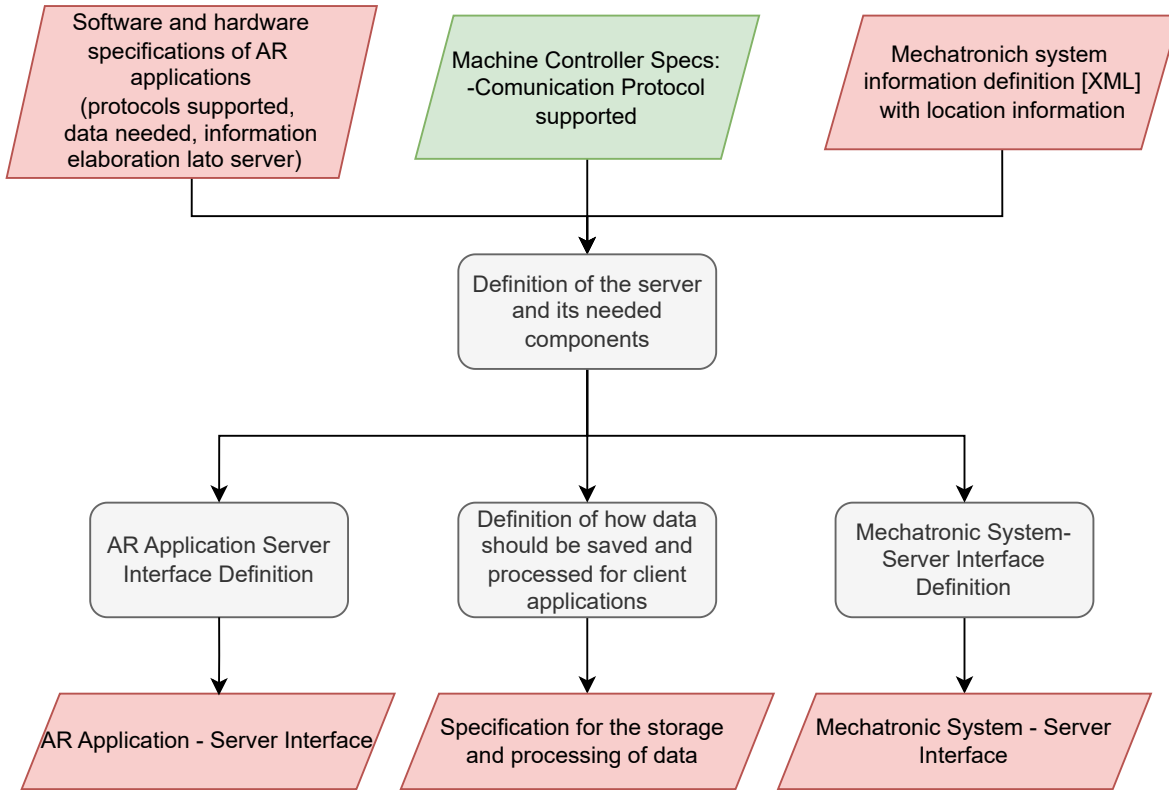


Figure 11: Server design phase

### 3.4 AR Application Development Model: Workflow Based on the Information Model

At this point, following all the previous steps, the system architecture has been defined, and the design of all the components has been established through the procedure. The next step is therefore the realization of all the components clearly defined during the process. For the development of the AR

application for monitoring and controlling the mechatronic system, the framework proposes a model to standardize the development of the application and leverage the object-oriented information model defined for mechatronic systems.

As a first step, to understand the model that will be proposed, an overview of the system is provided through a temporal representation of the activities among the three primary components: the server, the AR application, and the machine controller. This activity diagram (UML), Figure 12, provides a dynamic view of the interactions and data flows, clarifying how these components collaborate to achieve real-time monitoring and control in a mechatronic system.

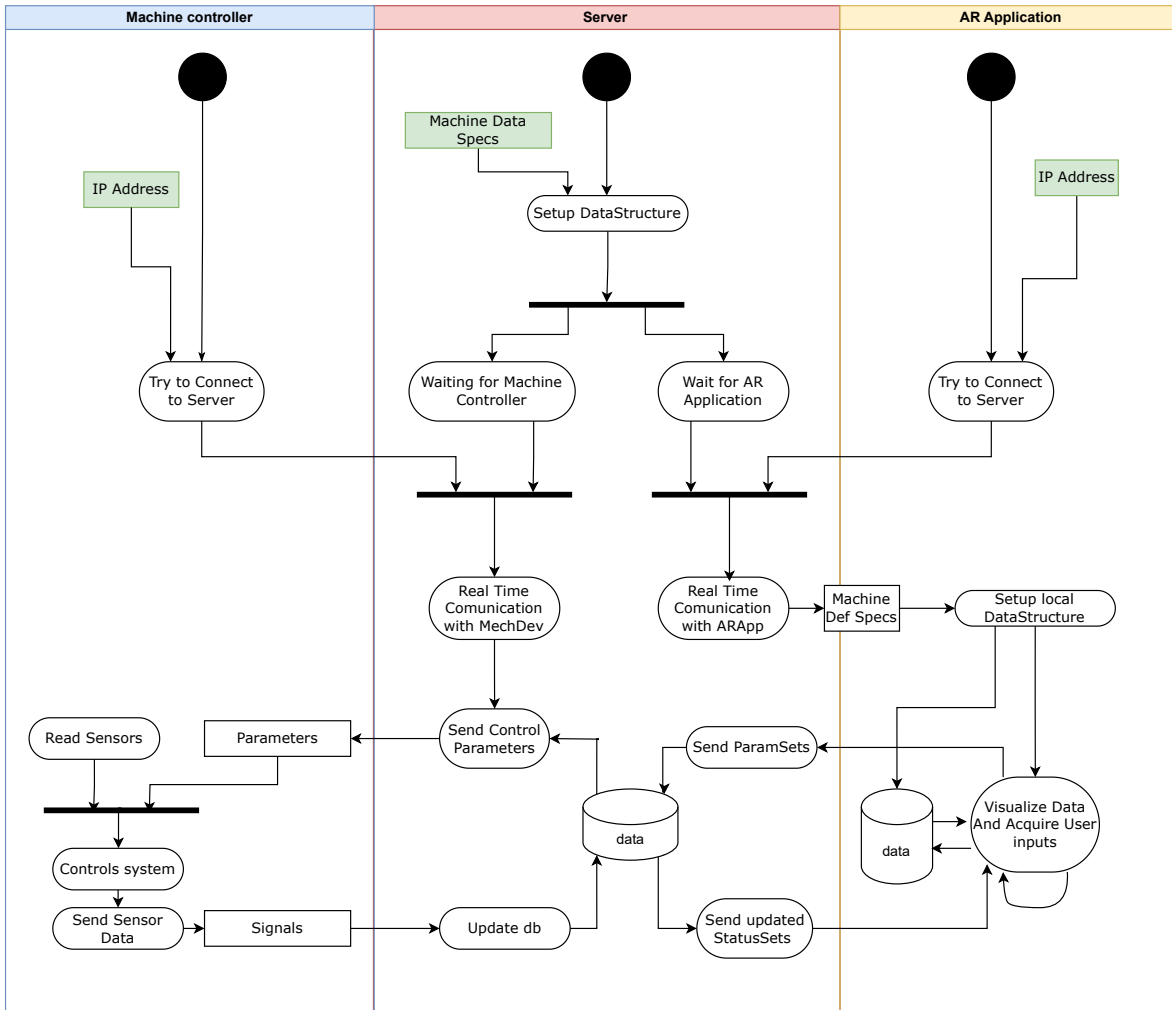


Figure 12: System activity diagram

The server assumes a central role in coordinating communication and data management between the AR application and the machine controller. Its workflow begins with setting up the data structure, based on the model proposed earlier. Once the structure is established, the server switches to a waiting state, waiting for connections from the AR application or the machine controller.

When the machine controller establishes a connection to the server via the IP address. Once connected, it starts transmitting sensor data and receiving control parameters in real time. This allows the

machine controller to perform its main functions, such as reading sensors, controlling system processes and sending signals to actuators or other components. Considering that the system controller already exists, the functionality to be integrated only concerns the functionality related to data exchange.

When the AR application connects, the server sends the mechatronic system information model so the local data structure of the AR application can be initialised. The AR application then begins its real-time operations, which include displaying data, which is always up-to-date, and capturing and sending user input.

A notable feature of this activity diagram is the real-time communication loop managed by the server, that continuously updates the database with data from the machine controller and the AR application. Both the AR application and the machine controller rely on the server as a hub for data exchange. This ensures that all components have synchronized and accurate information, enabling seamless operation.

The activity diagram also sheds light on the distinct but interdependent roles of each component. The AR application serves as the user-facing element, presenting data in a meaningful way and enabling interaction with the system. The machine controller, on the other hand, interfaces directly with the physical components of the mechatronic system, such as sensors and actuators, ensuring proper execution of operations. The server acts as a mediator, ensuring data consistency, synchronizing the activities of the AR application and machine controller, and maintaining the integrity of the system's data flow.

By presenting this activity diagram, the temporal flow of interactions is clarified, making it easier to identify potential bottlenecks, optimize processes, and ensure scalability.

Focusing on the AR application, the activity diagram presents a very general version of the activities it must perform. Starting from this simplistic view, in Figure 13, the detailed model proposed for the development of the application is shown.

The flowchart outlines a structured approach to developing an AR application that utilizes class-based modeling of a mechatronic system's information for monitoring and control purposes. The workflow is described in the following steps:

**1. Server Connection Initialization:**

- The process starts by attempting to connect to the server hosting the system's information model.
- This connection is essential as it provides access to the data needed for tracking and monitoring tasks.

**2. Target Tracking Procedure:**

- Upon successful connection, the application initiates a tracking procedure to locate the target in the physical environment.
- If the target is found, the application has the origin coordinates of the 3D model, needed to locate all the components; otherwise, the system handles the error.

**3. Exploring the Node Tree:**

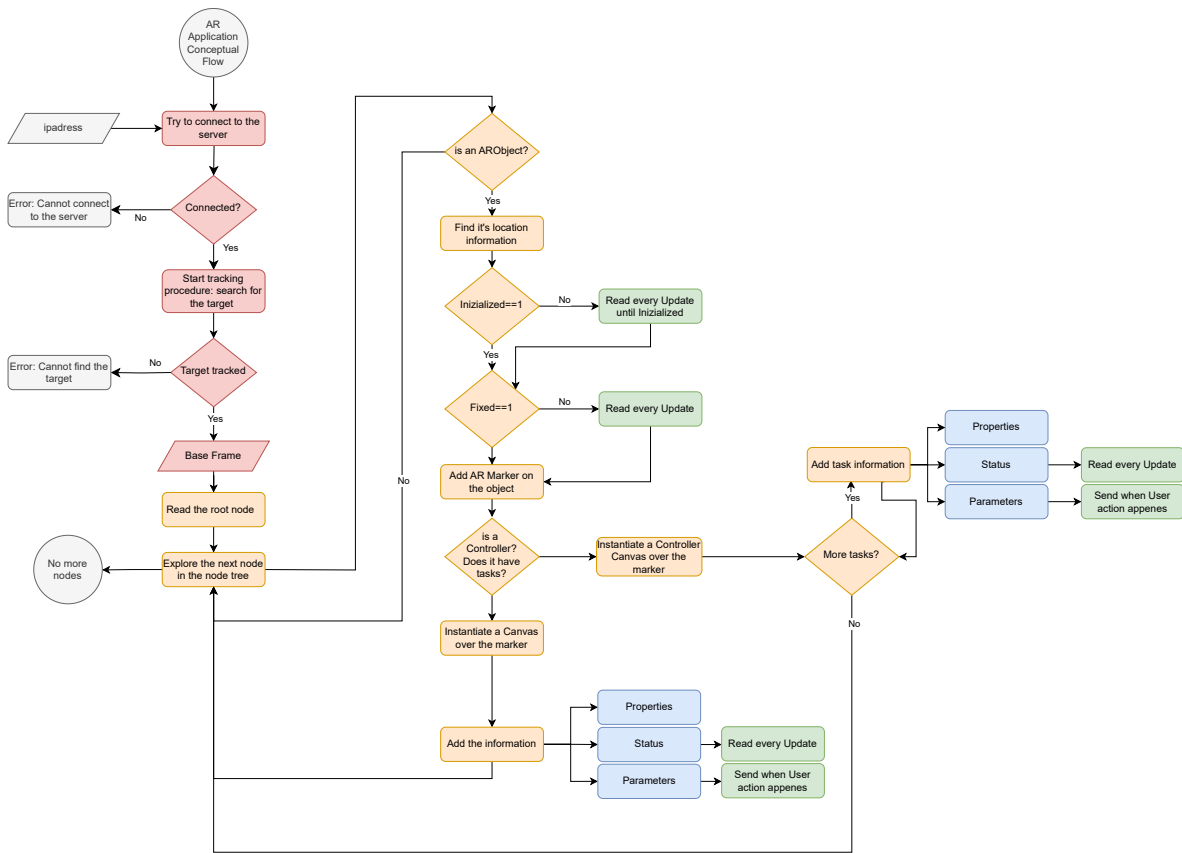


Figure 13: AR Application detailed flow

- The application begins traversing the node tree of the system's information model from the root node, the ARMechatronicSystem class.

#### 4. AR Object Determination:

- For each node, the application checks whether it represents an AR object.
- If true, additional processing steps are triggered, such as retrieving the object's location information and adding UI Canvas.

#### 5. Dynamic Properties Management:

- Depending on the node's properties:
  - If not initialized (`Inizialized == 0`), further checks are deferred to a later time.
  - If flagged as fixed (`Fixed == 1`), updates are read continuously, ensuring real-time movements reflection in the AR representation.
  - An AR marker is added to visually represent the object in the AR environment, using location information.

#### 6. UI Canvas Integration:

- For nodes identified as AR objects:
  - A canvas is instantiated above the marker to provide an interactive user interface displaying:

- \* **Properties:** Descriptive attributes of the object.
- \* **Status:** Current operational state or conditions.
- \* **Parameters:** Configurable parameters of the object.
- For nodes identified as Controllers:
  - The controller canvas is instantiated above the marker to provide an interactive user interface displaying all the tasks, with:
    - \* **Properties:** Descriptive attributes of the task.
    - \* **Status:** Current operational state or variables to be monitored.
    - \* **Parameters:** Control variables of the task.

Data updates are triggered by user actions (parameters) and periodic updates (states).

### 7. Termination of Node Exploration:

- The procedure ends when all nodes in the information model have been processed.
- This ensures the entire system is effectively mapped within the AR application.
- The terminal node refers to the setup phase, while the cyclic tasks (green rounded rectangles) are executed as long as the application remains running.

The proposed model allows the creation of an application that autonomously, with only the knowledge of the server's IP address, connects to it, downloads all the data, and by exploring them using a graph structure, adds markers to the components and to each one a user interface (defined by the user) with all the data. With this configuration, it is also possible to add a special canvas in case a controller with associated tasks is present, if the application aims to allow the control and monitoring of processes as well. Once the software application is developed for the first time, it will then be adaptable to all systems that implement the same information modeling.

## 3.5 Conclusions

This chapter presented an innovative framework for integrating mechatronic systems with augmented reality (AR) applications. Building on an analysis of the limitations identified in the literature, the framework provides a structured methodology to manage the challenges related to the use of augmented reality in industrial contexts.

The main aspects of the framework are summarized as follows:

- **Architecture definition:** Introducing a server as an intermediary between mechatronic systems and AR applications ensures modularity, flexibility, and scalability. This approach simplifies data management and facilitates integration across heterogeneous technologies.
- **Information modeling:** An object-oriented paradigm was adopted to define a generic model for representing components, states, and parameters of mechatronic systems. The model is enriched with spatial data to enable seamless integration into AR environments.
- **Design procedure:** A structured methodology was proposed to develop hardware and software specifications, model the information of mechatronic systems, and define the necessary communication interfaces.

- AR application development: The proposed model allows for the creation of reusable AR applications capable of dynamically adapting to any system that implements the defined information model.

In the next chapter, the proposed framework will be validated by applying it to the development of a monitoring and control application for a Cartesian robot (the mechatronic system) using augmented reality. This application will demonstrate the effectiveness of the framework in a real-world scenario.

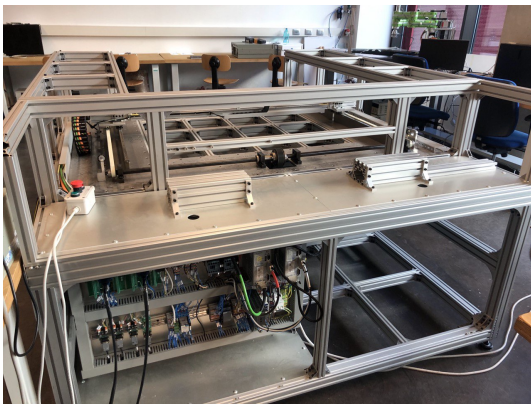
## 4 Applying the framework to a case study

### 4.1 Introduction

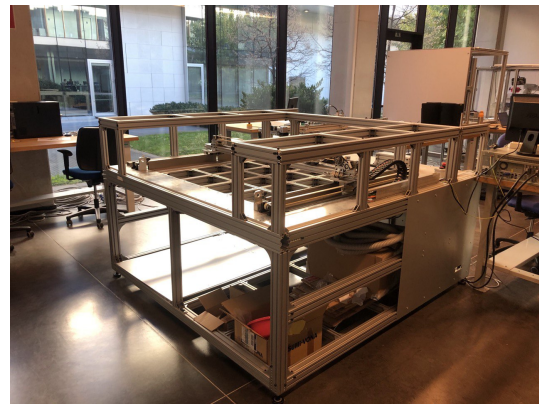
In the Chapter 3, the framework proposed in this thesis was introduced to address the challenges of integrating mechatronic systems with applications based on Augmented Reality (AR). The framework was divided into three conceptual parts:

- The definition of the target system architecture,
- The procedure for defining the software and hardware specifications of the aforementioned architecture, and
- The definition of the software structure of the involved systems, with a particular focus on proposing a reference schema for creating an AR application that is both effective and applicable to any system whose information is modeled as defined by the framework.

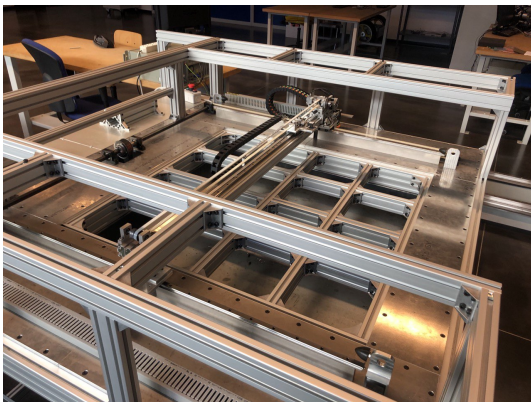
In this chapter, the proposed framework will be validated by applying each step of the procedure. The chosen mechatronic system for validation is a Cartesian robot located in the Mechatronics and Mechanical Dynamics Laboratory at the University of Bergamo.



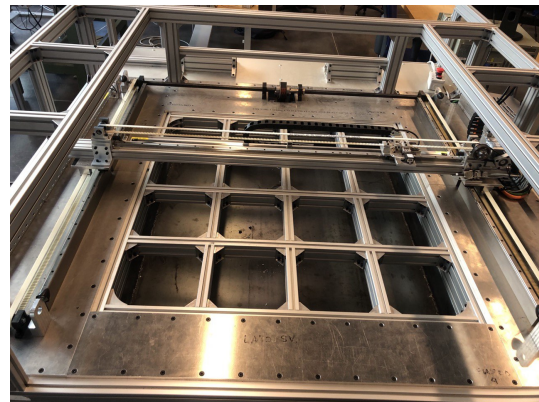
(a) Front view



(b) Side view



(c) Axes detail



(d) Axes detail

Figure 14: Cartesian robot in the Mechatronics and Mechanical Dynamics Laboratory

Figure 14 depicts the robot. Specifically, the front view (a) shows the control panel housing the electronic components, while images (c) and (d) highlight the two axes in detail. By convention, the

X-axis moves the End Effector, while the Y-axis moves the X-axis. Figure 15 shows a 3D model of the Cartesian robot with its axes exploded along the Z-axis. This view reveals the fixed structure of the Cartesian robot, the Y-axis just above it with mounting platforms for the X-axis, and finally, the X-axis itself, which moves a platform designed to hold the End Effector.

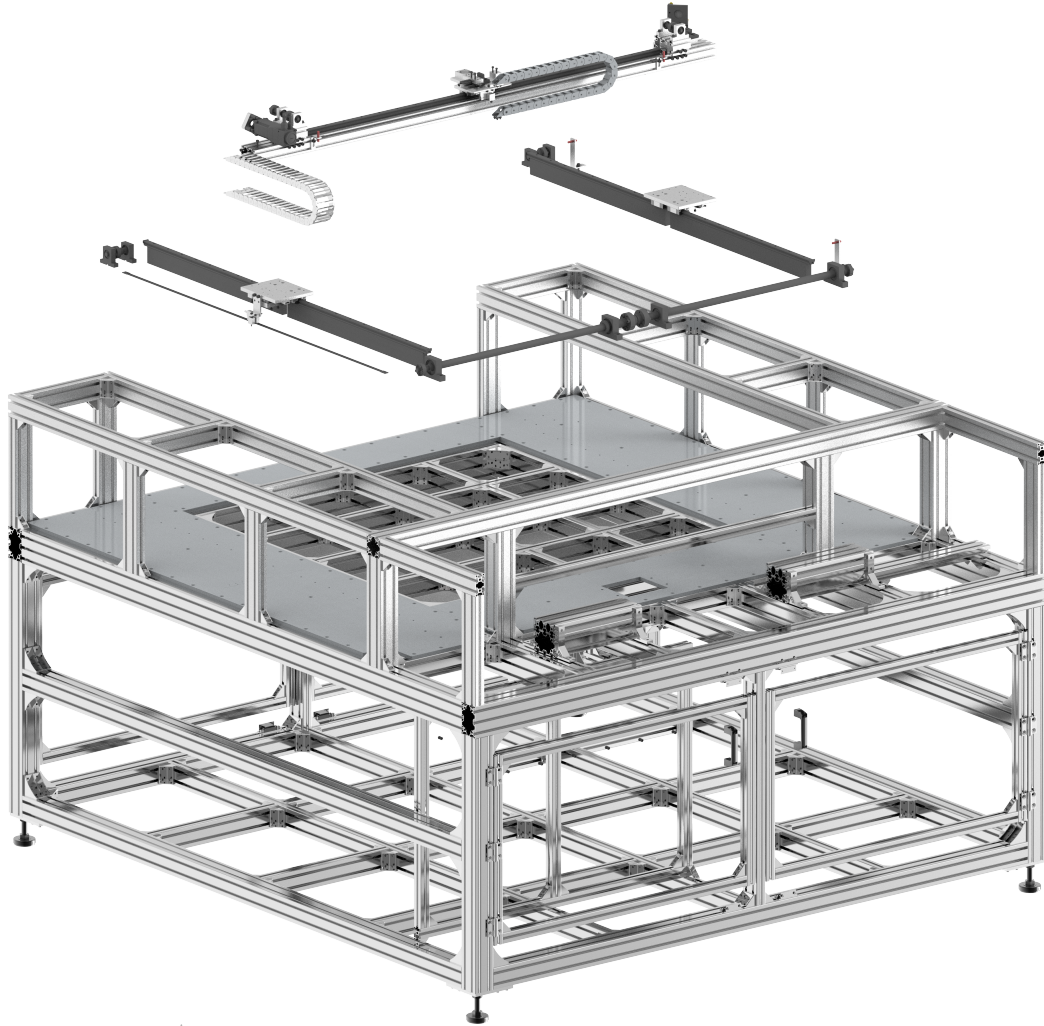


Figure 15: Cartesian robot's 3D model

The following sections describe the implementation of the framework. Specifically:

- The first one, explains the specification definition procedure, starting with the AR application by outlining its goals, functional requirements, and non-functional requirements.
- Next, the representation of the Cartesian robot's information, as the chosen mechatronic system, is presented, following the proposed information model.
- Lastly, the intermediary between the two systems, the server, is defined.

Upon completion of the procedure, all necessary information for system implementation will be available, allowing the realization phase to begin. This phase will focus on developing the system's three main components, following an order opposite to that of the specification definition procedure. Since

the server serves as the interconnection medium between the two systems, its specifications must be defined only after specifying the other two components. However, during the implementation phase, the server is the first component to be developed, enabling the subsequent creation of the AR application and the modifications required to make the mechatronic system's data available.

## 4.2 Design Process

### 4.2.1 AR Application design phase

Category	Requirement Description
Use Context	<p><b>Purpose of the application:</b> The AR application must support monitoring and control of a cartesian robot.</p> <p><b>Indoor or outdoor environment:</b> The application should be designed for indoor industrial environments.</p> <p><b>Duration of use per session:</b> The device should support at least 4 hours of continuous usage for practical, long-term monitoring.</p> <p><b>Compatibility:</b> The application does not have to be strictly hardware-related. It must be able to run on different devices.</p>
Key Features	<p><b>Type of AR:</b> The application should be markerless, meaning it does not rely on physical markers for tracking, using 3D model recognition.</p> <p><b>Tracking precision requirements:</b> A high level of tracking precision is required to interact with and monitor the mechatronic system in real-time.</p> <p><b>Visualization of 3D models:</b> The device should have a high-performance GPU to render and display complex 3D models smoothly and clearly.</p>
User Interaction	<p><b>Interaction methods:</b> The application must support hand gestures and use of an external controller for diverse user input methods.</p>
Integration	<p><b>Communication with middleware or server:</b> The system should support Wi-Fi 6, to facilitate wireless communication with servers for data exchange.</p>
User Ergonomics	<p><b>Comfort for prolonged use:</b> The device should be lightweight and comfortable for long periods of use, with adjustable features to suit various users.</p>
Robustness	<p><b>Stability of tracking in dynamic environments:</b> The AR system must maintain stable tracking even in dynamic or moving environments, allowing for accurate monitoring of the mechatronic system.</p>
Target Audience	<p><b>Expertise level of the end-user:</b> The user interface should be simple and intuitive, designed for users who may not be experts in AR or the underlying technology.</p> <p><b>Simplicity of the user interface:</b> The application must have an easy-to-navigate interface.</p>

Table 4: Definition of AR Application Requirements for Mechatronic System Monitoring

This initial section describes the first step of the procedure concerning the design of the AR application, which runs concurrently with the definition of the information model for the mechatronic system, covered in the next section. As detailed in the chapter that explains this phase of the procedure [3.3], this step requires defining both functional and non-functional requirements. These requirements, once translated into specifications, establish the constraints that must guide the choice of software and hardware for the AR application's development.

Requirement	Technical Specification	Magic Leap 2 Features
<b>Use Context</b>	<b>**Purpose of the application**</b> : Head-mounted device for immersive AR application, leaving hands free for interaction with the mechatronic system.	Magic Leap 2 is a head-mounted AR device that offers immersive experiences while allowing hands-free interaction with the environment.
	<b>**Indoor or outdoor environment**</b> : Support for operation in indoor settings.	Magic Leap2 ensures functionality in indoor industrial environments.
	<b>**Duration of use per session**</b> : At least 4 hours of continuous use.	Up to 3-4 hours of battery life on a single charge.
	<b>**Compatibility**</b> : Must be compliant with the OpenXR standard .	Starting with OS version 1.5.0, Magic Leap 2 supports OpenXR.
<b>Key Features</b>	<b>**Type of AR**</b> : Markerless AR application using 3D model tracking.	Depth camera for markerless AR and spatial mapping, enabling 3D model tracking without markers.
	<b>**Tracking precision requirements**</b> : High-precision tracking of 3D models.	Magic Leap 2 features high-resolution depth sensors and advanced SLAM technology for precise tracking in real-time.
	<b>**Visualization of 3D models**</b> : Smooth rendering of complex 3D models.	GPU and hardware optimized for high-performance 3D rendering, enabling smooth visualization of complex models.
<b>User Interaction</b>	<b>**Interaction methods**</b> : Hand gesture recognition and controller support.	Magic Leap 2 supports hand gestures with advanced recognition and includes a controller for additional input.
<b>Integration</b>	<b>**Communication with middleware or server**</b> : Wi-Fi communication.	Magic Leap 2 supports Wi-Fi 6 for fast and reliable communication with external servers and systems.
<b>User Ergonomics</b>	<b>**Comfort for prolonged use**</b> : Lightweight and comfortable design.	Magic Leap 2 weighs less than 250g and features adjustable head straps and a separate compute pack for enhanced comfort.
<b>Robustness</b>	<b>**Stability of tracking in dynamic environments**</b> : Stable tracking in motion.	Features advanced motion compensation and sensor fusion, including accelerometers and gyroscopes, to maintain tracking stability even in dynamic conditions.
<b>Target Audience</b>	<b>**Expertise level of the end-user**</b> : Simple and intuitive interface.	Magic Leap 2 includes an easy-to-use interface with onboarding tutorials and a guided setup process for non-expert users.
	<b>**Simplicity of the user interface**</b> : Intuitive and beginner-friendly interface.	The user interface is designed to be intuitive with touch-based interaction and easy access to key features, suitable for novice users.

Table 5: Requisite AR Application and Magic Leap 2 Specifications

Referring to the guidelines table for requirement definition, Table 2, a specific table for this case study’s requirements has been created, Table 4. The application’s purpose aligns with the overall framework’s goal—monitoring and control a mechatronic system—but additional details are provided about how this purpose must be achieved.

Among the most impactful requirements for both software and hardware selection are the type of AR technology (markerless, in this case), the interaction methods (using a controller or hand gestures), and compatibility. Specifically, the application must be designed to work across multiple devices with minimal modifications required for adaptation. This requirement is essential for creating reusable and adaptable software and has broader implications, which will be explored in later sections.

Once the requirements were defined, they were translated into technical specifications. The first consideration was whether to use a head-mounted or handheld device. In this case, the goal was to develop a highly immersive application that would allow users to keep their hands free for interacting with information, either through hand gestures or using a controller. As such, a headset was chosen. Given the intent to enable interaction via hands or a controller, this method of interaction was established as a key specification.

To develop a markerless application, it was deemed essential for the device to be equipped with a depth camera, enabling the effective management of 3D models. Connectivity to the server is ensured wirelessly, with support for Wi-Fi 6 to guarantee high-speed and reliable communication.

All the specifications are summarized in Table 5, where the final column highlights the features of the selected headset. This serves to validate the considerations made during the decision-making process.



Figure 16: Magic Leap 2 Headset, Compute Pack and Controller

The Magic Leap 2 headset, shown in Figure 16, is an advanced augmented reality (AR) device designed specifically for enterprise and professional use. Building on the features of its predecessor, it offers enhanced performance and capabilities. One of its standout features is the introduction of Dynamic Dimming technology, which minimizes light interference in bright environments, ensuring that virtual content remains vivid and legible even under challenging lighting conditions. The device also features a wide field of view, illustrated in Figure 17, measuring 70 degrees diagonally, providing a more immersive AR experience compared to many other headsets on the market.

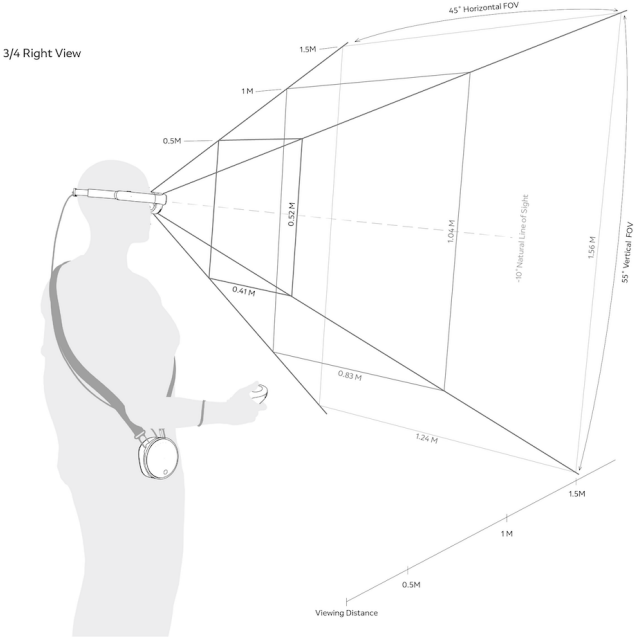


Figure 17: Magic Leap2 FOV

Ergonomics play a central role in the design of the Magic Leap 2. The headset is lightweight and carefully balanced to ensure comfort during extended periods of use. Unlike traditional all-in-one headsets, it features a compute pack that is worn on the waist, reducing the weight on the user’s head and improving mobility. This compute pack houses an AMD Quad-Core Zen 2 processor, along with 16GB of RAM and 128GB of storage, enabling the device to handle complex applications with ease.

In terms of interaction and environmental awareness, the Magic Leap 2 excels with its advanced spatial mapping capabilities. It also includes both hand and eye tracking, allowing users to interact naturally with digital objects. The headset supports realistic occlusion, ensuring virtual objects blend seamlessly with the physical world, enhancing the realism of the AR experience.

For display, it delivers high-resolution visuals, complemented by spatial audio for an immersive auditory experience. Connectivity options include Wi-Fi 6 and Bluetooth 5.2, ensuring reliable and fast communication with other devices. The battery, housed in the compute pack, provides several hours of uninterrupted use, meeting the demands of professional workflows.

The Magic Leap 2 is particularly well-suited for enterprise use, offering compatibility with OpenXR and providing a robust SDK for developers to create tailored applications. The compatibility with

OpenXR fulfills the previously mentioned requirement for cross-platform support (\*\*Compatibility\*\*). OpenXR is an open standard developed by the Khronos Group to simplify the development of virtual reality (VR) and augmented reality (AR) applications across a variety of platforms and devices. By providing a single API, it enables developers to create software compatible with many headsets and systems without needing to adapt the code for each specific piece of hardware.

One of the key advantages of OpenXR is its cross-platform compatibility, allowing applications to function seamlessly on devices such as Meta Quest, HoloLens, SteamVR, and Magic Leap 2, which announced its adoption of the standard in 2022. This standard reduces development complexity by eliminating the need for proprietary SDKs for each platform, resulting in significant time and cost savings. Additionally, as an open standard, OpenXR fosters innovation and collaboration within a global ecosystem, offering a shared foundation for AR/VR application development. In the following section, dedicated to the practical implementation of the application, the implications of using OpenXR for development will be explored in greater detail.

The main technical features of the Magic Leap 2 headset are summarized in Table 6.

Table 6: Technical Specifications of Magic Leap 2

<b>Headset</b>	
<b>Display</b>	2x Micro-LED displays with dynamic dimming, FOV 70° diagonal
<b>Resolution</b>	1440 x 1760 pixels per eye
<b>Refresh Rate</b>	120 Hz
<b>Tracking</b>	Eye tracking, hand tracking, 6DoF positional tracking
<b>Sensors</b>	4 cameras for environment tracking, 1 depth sensor, 1 RGB camera (12.6 MP)
<b>Audio</b>	Built-in spatial audio speakers, 3.5mm audio jack
<b>Weight</b>	260 grams
<b>Controller</b>	
<b>Design</b>	Ergonomic, single-handed controller
<b>Tracking</b>	6DoF tracking with high precision
<b>Input Options</b>	Touchpad, trigger, bumper, and home buttons
<b>Battery Life</b>	Up to 7.5 hours of continuous use
<b>Connectivity</b>	Bluetooth Low Energy (BLE)
<b>Haptics</b>	Advanced haptic feedback
<b>Compute Pack</b>	
<b>Processor</b>	AMD Quad-core CPU, dedicated GPU
<b>RAM</b>	16 GB LPDDR5
<b>Storage</b>	128 GB internal storage
<b>Connectivity</b>	Wi-Fi 6, Bluetooth 5.2, USB-C
<b>Battery Life</b>	3.5 hours of continuous use
<b>Weight</b>	420 grams

Another reason for choosing this headset, beyond its technical requirements, relates to the state of the art regarding the use of Magic Leap 2 in industrial contexts. An analysis of related literature reveals that most reported applications pertain to the medical field. There is, in fact, no applied research addressing its use for the interaction with mechatronic systems.

Magic Leap has been utilized in non-medical contexts, such as visualization, museum experiences, and education. One study explored its application for map visualization [78], demonstrating the potential of AR to enhance user interaction with spatial data. Another study examined the intended and

realized experiences of head-mounted AR in a museum context. It compared user experiences with VR and AR versions of the immersive "Virtual Veronese" gallery, measuring factors like enjoyment, presence, and engagement [79]. The study at the National Gallery in London highlighted how AR can enrich cultural experiences [80].

In terms of mixed reality (MR) headsets, Magic Leap has been compared to other devices such as the Microsoft HoloLens 2. In this comparison, Magic Leap was preferred by users, particularly for its superior visualization quality and ease of interaction with 3D content, despite no significant differences in frame rate between the two devices [81].

In the medical field, however, the majority of contributions are concentrated. For example, in [82], Magic Leap was evaluated for the development of immersive environments, such as BioLumin, a system enabling users to virtually shrink to microscopic levels to navigate and annotate 3D reconstructed images.

Magic Leap's applications also extend to surgical training and planning. For instance, it has been used in preoperative evaluation to assist with planning surgeries, including 3D rendering for pedicle screw placements [83, 84]. Augmented reality has also been employed during intraoral scans to improve accuracy and efficiency, such as in scanning fully dentate mandibular arches [85]. Moreover, a novel technique for external approaches to the frontal sinus was explored using holographic AR applications [86].

In rehabilitation, particularly for patients with Parkinson's disease, Magic Leap has facilitated home-based exergaming interventions aimed at improving gait and balance. These AR glasses have shown promise as an effective therapeutic tool [87, 88, 89].

In medical education, Magic Leap has been integrated into training programs such as SkillsLab+, which utilizes AR and haptic feedback for medical training [90]. It has also been employed for simulation-based learning in neurosurgery, offering realistic training experiences in safe environments [91]. Additionally, its suitability for anatomy training has been tested, comparing Magic Leap to a semi-immersive VR setup [92]. Multiplayer AR has also been proposed as a valuable tool for remote medical education, particularly for communication skills during medical crises [93].

Despite the growing interest in Magic Leap, there is still a limited number of studies directly citing its use. Most of the research focuses on medical applications, particularly in surgery, training, and rehabilitation [94]. This limited body of work underscores the potential for further exploration and development of Magic Leap in other sectors, as exemplified in this study, which addresses the human machine interaction in industrial contexts.

#### **4.2.2 Mechatronic system information modelling phase**

Once the specifications for the AR application have been defined, it is possible to proceed with the phase focused on identifying the information related to the mechatronic system that will be monitored and controlled. It is worth noting that these two phases are concurrent and can be addressed independently, thanks to the presence of an intermediate server that decouples the interconnection challenges between the systems. For implementing the proposed information model within this framework, the

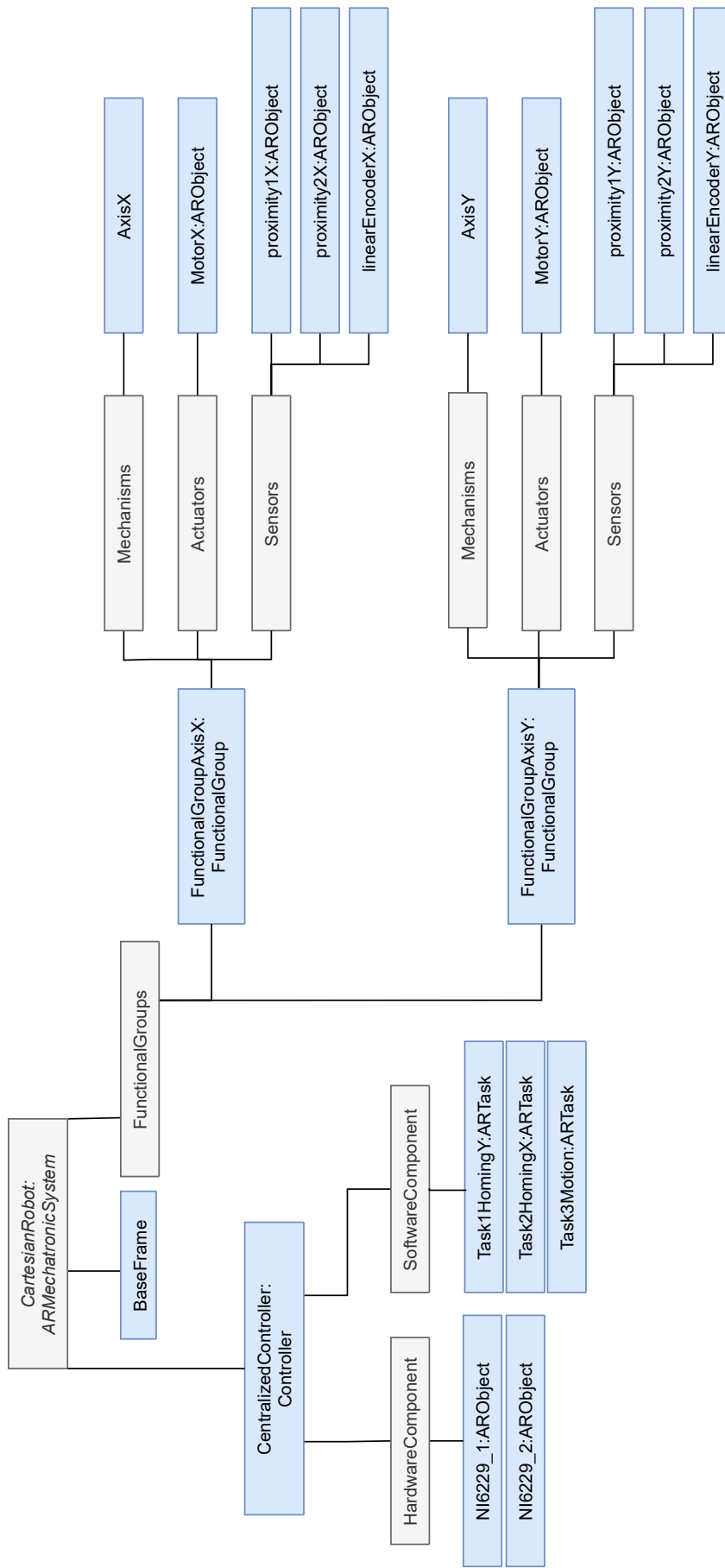


Figure 18: Application of the information model to the case study

model depicted in Figure 10 will be taken as a reference, which is thoroughly explained in Section 3.3.2. This model will be applied to the previously introduced Cartesian robot, detailing its components and the relationships between them.

The resulting model is illustrated in Figure 18.

Analyzing the diagram, we can observe the definition of the CartesianRobot class, where the notation : *ARMechatronicSystem* indicates the implementation of a class previously defined within the proposed information model. The system defined in this way includes a BaseFrame, which represents the origin coordinate system; a CentralizedController, so named because it is shared across all system resources; and two FunctionalGroups.

The controller contains two hardware components, specifically National Instruments acquisition boards, and three software components, corresponding to the tasks executed by the mechatronic system. The two functional groups represent the two axes of the Cartesian robot, or more generally, all the components that characterize them and enable their controlled movement: motors, sensors, and mechanisms. All components indicated as : *ARObject* and *ARTask* will subsequently provide information accessible to the AR application.

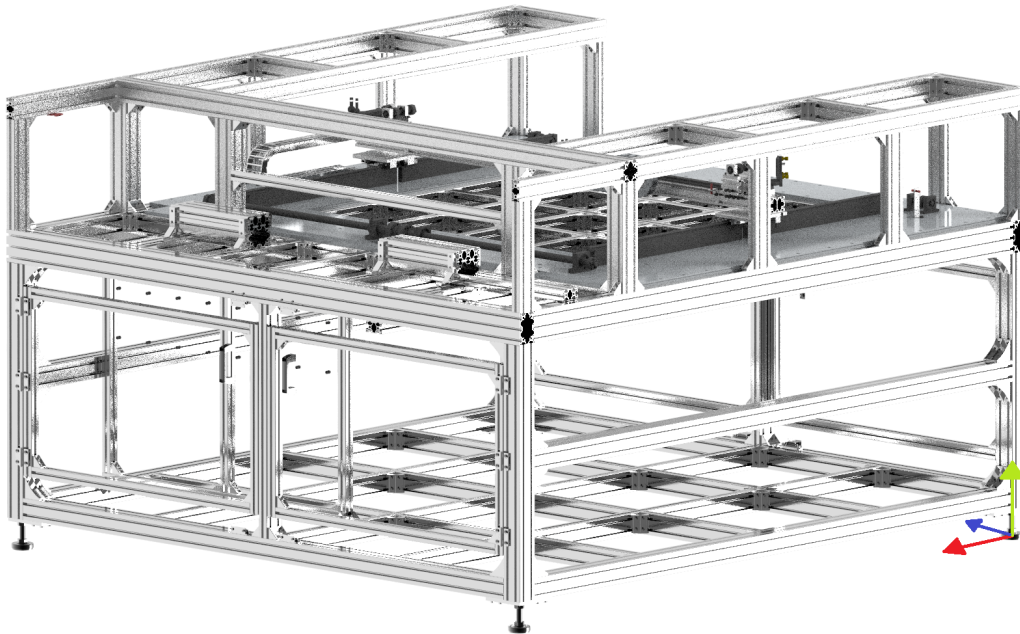


Figure 19: 3D model with Origin coordinates

Once the data structure was defined, it was enriched with all the localization information for the components. Starting from the 3D model of the Cartesian robot, the components were identified, and their coordinates were derived relative to the system's BaseFrame.

In this specific case, the origin of the coordinate system, or BaseFrame, corresponds to the rear-right support foot, as indicated in Figure 19. Specifically, the Y-axis (green) points upward, the X-axis (red) points toward the robot's front, and the Z-axis (blue) extends toward its left side.

As an example, Figure 20 highlights certain components on the 3D model whose positions were determined, including the limit switches for both axes and the linear encoders. Components related to the X-axis are highlighted in blue, while those related to the Y-axis are shown in red.

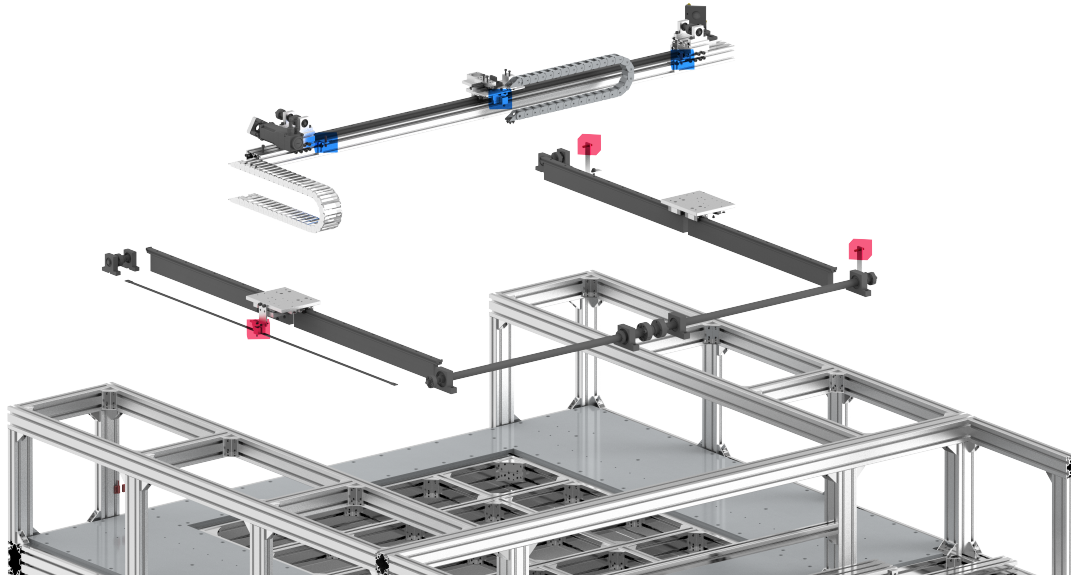


Figure 20: 3D model with marker on some sensors

All components defined as ARObjct were characterized based on the previously proposed model, categorizing them into properties (intrinsic and functional), state, parameters, and localization information. The tables presented below detail all components and the corresponding data in this structured format.

Each table will be briefly discussed, with particular focus on the functional properties to highlight their utility in establishing relationships between components. Additionally, the localization information will be examined, providing justification for the attributes fixed and initialized.

The Table 7 provides a detailed overview of the motors driving the X and Y axes of the Cartesian robot, organizing the information into status, intrinsic and functional properties, and location data.

For both MotorX and MotorY, the status indicates the possible states of the inverter's state machine. These states define the operational conditions of the drive and its interaction with the motor. The possible states include:

- NOT READY TO SWITCH ON: Low-level power is applied, and the drive is initializing or running self-tests. If a brake is present, it must be applied. The drive function is disabled.
- SWITCH ON DISABLED: Initialization is complete, and parameters are set up and modifiable. High voltage is not applied, typically for safety reasons. The drive function is disabled.

Table 7: X and Y axis motor data

<b>MotorX</b>		<b>MotorY</b>	
<b>Status Set</b>		<b>Status Set</b>	
Status	X	Status	X
<b>Intrinsic Properties</b>		<b>Intrinsic Properties</b>	
Manufacturer	Parker	Manufacturer	Mavilor
Model	SMB	Model	BLS-073
Product Code	SMB42600 4559S064	Product Code	BS073A,00,3105,00
Stall Current	0.78	Stall Torque	2.7
KtNmA	0.455	KtNmA	0.69
<b>Functional Properties</b>		<b>Functional Properties</b>	
Moves	AxisX	Moves	AxisY
<b>Location Information</b>		<b>Location Information</b>	
x	-	x	-1.4308
y	1.0428	y	0.7922
z	1.6299	z	0.8173
Fixed	0	Fixed	1
Initialized	0	Initialized	1

- **READY TO SWITCH ON:** High voltage can be applied, parameters can be changed, but the drive function remains disabled.
- **SWITCHED ON:** High voltage is applied, and the power amplifier is ready. Parameters are modifiable, but the drive function is still disabled.
- **OPERATION ENABLE:** No faults are detected. The drive function is enabled, power is applied to the motor, and parameters remain modifiable. This state represents normal operation.
- **QUICK STOP ACTIVE:** The quick stop function is executing. Power remains applied to the motor, and parameters are modifiable.
- **FAULT REACTION ACTIVE:** A fault has occurred, and the quick stop function is active. Power is applied to the motor, and parameters can be changed.
- **FAULT:** A fault has occurred, and the drive function is disabled. High voltage application depends on the specific use case.

Both motors have well-defined intrinsic properties: MotorX, manufactured by Parker, is an SMB model with a product code of SMB42600 4559S064, a stall current of 0.78 A, and a torque constant (KtNmA) of 0.455. MotorY, manufactured by Mavilor, is a BLS-073 model with a product code of BS073A,00,3105,00, a stall torque of 2.7 Nm, a torque constant (KtNmA) of 0.69, and a back-emf constant (KeVsrd) of 0.4. These properties highlight the motors' respective physical and electrical characteristics.

Functionally, MotorX is associated with AxisX, while MotorY is linked to AxisY, indicating their roles in controlling the corresponding axes. Regarding location information, MotorX is neither fixed nor initialized (Fixed: 0, Initialized: 0). This is because MotorX is constrained to AxisX, which moves along AxisY. It cannot be fixed due to its dependency on AxisY for motion. Furthermore, MotorX remains uninitialized until AxisY completes its zeroing phase, enabling the linear encoder on AxisX

to be reset.

In contrast, MotorY is both fixed and initialized (Fixed: 1, Initialized: 1). This configuration reflects its static positioning and readiness within the system.

The functional properties emphasize the motors' relationships with their respective axes, while the location information, including the Fixed and Initialized attributes, clarifies their spatial positioning, dynamic dependencies, and readiness within the robotic system.

The Table 8 provides data for the linear encoders used to measure the position of the X and Y axes of the Cartesian robot. Both encoders are not fixed (Fixed: 0) nor initialized (Initialized: 0) at the system startup, reflecting their dependency on a zeroing phase to establish their reference positions. This is typical for systems where absolute positioning is not predefined, and the encoders require a reference or zeroing operation to determine their position. Despite this limitation, the encoders' definition ensures that at least the Y-coordinate is known at all times since they operate within a plane. For LinearEncoderY, only the X-coordinate varies, in contrast, LinearEncoderX has both X and Y coordinates that vary due to the coupling between the axes, with its position influenced by the motion along both X and Y axes.

Table 8: Linear Encoders data

<b>LinearEncoderX</b>		<b>LinearEncoderY</b>	
<b>Status Set</b>		<b>Status Set</b>	
Status	X	Status	X
Actual Value	-	Actual Value	-
<b>Intrinsic Properties</b>		<b>Intrinsic Properties</b>	
Manufacturer	Lika	Manufacturer	Lika
Model (Magnetic Tape)	MT50-2-100-1	Model (Magnetic Tape)	MT50-2-100-1
Model (Sensor)	SME51-L-1-5-N-L5-J	Model (Sensor)	SME51-L-1-5-N-L5-J
Resolution	5 micrometers	Resolution	5 micrometers
Length Meter	2 m	Length Meter	2 m
Supply Voltage	24 V	Supply Voltage	24 V
<b>Functional Properties</b>		<b>Functional Properties</b>	
Measures	AxisX ActualPos	Measures	AxisY ActualPos
<b>Location Information</b>		<b>Location Information</b>	
x	-	x	-
y	1.0037	y	0.896
z	-	z	1.6623
Fixed	0	Fixed	0
Initialized	0	Initialized	0

The Table 9 provides information about the two National Instruments PCI 6229 I/O boards, NI6229\_1 and NI6229\_2, which play a crucial role in interfacing with the sensors and encoders of the system. Both boards share similar specifications and are fully initialized (Initialized: True) and fixed (Fixed: True). Their physical positions within the system are specified by their respective coordinates.

NI6229\_1 is connected to the Proximity1X, Proximity2X, and LinearEncoderX, enabling it to process analog and digital signals related to the X-axis components. Similarly, NI6229\_2 is linked to Proximity1Y, Proximity2Y, and LinearEncoderY, handling signals associated with the Y-axis. Both the NI boards have no status information.

Table 9: National Instruments PCI 6229 I/O Boards data

<b>NI6229_1</b>		<b>NI6229_2</b>	
<b>Intrinsic Properties</b>		<b>Intrinsic Properties</b>	
Manufacturer	National Instrument	Manufacturer	National Instrument
Model	PCI 6229	Model	PCI 6229
Analog Inputs	32	Analog Inputs	32
ADC Resolution	16 bits	ADC Resolution	16 bits
Analog Outputs	4	Analog Outputs	4
PFI Digital I/Os	48	PFI Digital I/Os	48
<b>Functional Properties</b>		<b>Functional Properties</b>	
Connections	Proximity1X, Proximity2X, LinearEncoderX	Connections	Proximity1Y, Proximity2Y, LinearEncoderY
<b>Location Information</b>		<b>Location Information</b>	
x	-1.7284	x	-1.7284
y	0.611	y	0.611
z	1.664	z	1.5776
Fixed	True	Fixed	True
Initialized	True	Initialized	True

Tables 10 and 11 provide detailed data on the proximity sensors mounted on the Y-axis and X-axis, respectively. Analyzing the data reveals significant differences between the two sets of sensors, particularly regarding their fixed position and initialization status.

The sensors mounted on the Y-axis, ProximitySensor1Y (Start Sensor) and ProximitySensor2Y (End Sensor), are both marked as Fixed and Initialized. This indicates that their position in space is predefined and known, eliminating the need for a homing process to initialize their positions. ProximitySensor1Y measures the left limit of the Y-axis (AxisYleftLimitSwitch), while ProximitySensor2Y measures the right limit of the Y-axis (AxisXrightLimitSwitch).

Table 10: Proximity Sensors Data (Axis Y)

<b>ProximitySensor1Y (Start Sensor)</b>		<b>ProximitySensor2Y (End Sensor)</b>	
<b>Status Set</b>		<b>Status Set</b>	
Status	X	Status	X
Actual Value	-	Actual Value	-
<b>Intrinsic Properties</b>		<b>Intrinsic Properties</b>	
Manufacturer	Assemtch Europe	Manufacturer	Assemtch Europe
Model	PTC 130/30	Model	PTC 130/30
Supply Voltage	24 V	Supply Voltage	24 V
<b>Functional Properties</b>		<b>Functional Properties</b>	
Measures	AxisYleftLimitSwitch	Measures	AxisXrightLimitSwitch
<b>Location Information</b>		<b>Location Information</b>	
x	-1.3286	x	-0.2
y	0.9554	y	0.9554
z	0.1652	z	0.1652
Fixed	1	Fixed	1
Initialized	1	Initialized	1

Table 11: Proximity Sensors Data (Axis X)

<b>ProximitySensor1X (Start Sensor)</b>		<b>ProximitySensor2X (End Sensor)</b>	
<b>Status Set</b>		<b>Status Set</b>	
Status	Working	Status	Working
Actual Value	0	Actual Value	0
<b>Intrinsic Properties</b>		<b>Intrinsic Properties</b>	
Manufacturer	Assemtech Europe	Manufacturer	Assemtech Europe
Model	PTC 130/30	Model	PTC 130/30
Supply Voltage	24 V	Supply Voltage	24 V
<b>Functional Properties</b>		<b>Functional Properties</b>	
Measures	AxisXleftLimitSwitch	Measures	AxisXrightLimitSwitch
<b>Location Information</b>		<b>Location Information</b>	
x	-	x	-
y	1.0037	y	1.0037
z	1.4942	z	0.3442
Fixed	0	Fixed	0
Initialized	0	Initialized	0

In contrast, the sensors mounted on the X-axis, ProximitySensor1X (Start Sensor) and ProximitySensor2X (End Sensor), are marked as not Fixed and not Initialized. This implies that their position is not predefined or fixed and must be determined through a homing process. This dependency can be attributed to the fact that the X-axis is mounted on the Y-axis, making the position of the X-axis sensors reliant on the Y-axis configuration. The coordinates of the Y and z axes are known, the position of the Y axis affects only the X coordinate. ProximitySensor1X measures the left limit (AxisXleftLimitSwitch), while ProximitySensor2X measures the right limit (AxisXrightLimitSwitch).

Finally, Table 12 presents the tasks performed by the Cartesian robot. Considering that the goal of the framework is to enable monitoring and control of the Cartesian robot through augmented reality, it becomes essential to have information not only about its components but also about the processes it executes.

Each task is characterized by its status, which can be used to monitor related variables, by parameters, which users can manipulate to control the process, and by properties. In this example, the functional properties of the tasks indicate the functional group responsible for them. This allows the user to monitor and identify any malfunctioning components connected to a failing task.

In this paragraph, it has been demonstrated how to apply the proposed information model to the investigated Cartesian robot. The nature of the model allows representing all the necessary information to bring the data of the mechatronic system into an augmented view while maintaining a simple and easily applicable structure. In the following paragraphs, it will be shown how the derived model will be implemented on the server side to make static information available to the AR application, as well as establishing communication channels to update dynamic information such as status and parameters.

Table 12: Tasks Data

Task	Details
<b>Task1HomingY</b>	<p><b>Status Set</b> Status: Idle-Working-Error-...</p> <p><b>Param Set</b> Start: 0-1 Stop: 0-1</p> <p><b>Intrinsic Properties</b> Description: Moving Y Axes to Home position</p> <p><b>Functional Properties</b> Performed by: FunctionalGroupAxisY</p>
<b>Task2HomingX</b>	<p><b>Status Set</b> Status: Idle-Working-Error-...</p> <p><b>Param Set</b> Start: 0-1 Stop: 0-1</p> <p><b>Intrinsic Properties</b> Description: Moving X Axes to Home position</p> <p><b>Functional Properties</b> Performed by: FunctionalGroupAxisX</p>
<b>Task3Motion</b>	<p><b>Status Set</b> Status: Idle-Working-Error-...</p> <p><b>Param Set</b> Start: 0-1 Stop: 0-1</p> <p><b>Intrinsic Properties</b> Description: Moving the EE</p> <p><b>Functional Properties</b> Performed by: FunctionalGroupAxisY, FunctionalGroupAxisX</p>

### 4.2.3 Server design phase

At this stage of the procedure, the information model of the Cartesian robot has been developed, and the specifications for the AR application have been defined. These details, combined with prior knowledge of the robot's existing control system, allow for the design of the server acting as an intermediary between the two systems. Among the design decisions to be made, as illustrated in Figure 11, are determining how to handle data consistently, efficiently, and securely, as well as how to interface with the two clients: the AR application and the robot controller.

The robot controller is an industrial PC running a program developed in Simulink Desktop Real-Time. This controller lacks wireless network capability and supports only wired connections. Among the Ethernet-based protocols natively supported are UDP and TCP. To manage this configuration, one option is to connect a network port to a switch-like device to enable wireless connectivity. Alternatively, the industrial computer can be directly connected to the server via cable, maximizing performance and utilizing the server as an edge device. For the AR application, the headset supports wireless connectivity, including Wi-Fi 6, and is compatible with all major protocols.

The solution adopted in this case relies on OPC UA, which, being based on the TCP/IP protocol, is supported by both devices. In addition to enabling data exchange between the two systems, OPC UA provides the capability to use complex information models based on the object-oriented paradigm, which aligns well with the data representation proposed by this framework.

<b>Aspect</b>	<b>OPC Classic</b>	<b>OPC UA</b>
<b>Architecture</b>	Client-Server based on COM/D-COM	Service-Oriented Architecture (SOA)
<b>Platform Dependency</b>	Tied to Windows (via COM/D-COM)	Platform-independent (supports Windows, Linux, macOS, and more)
<b>Communication Protocols</b>	Proprietary (DCOM)	Standardized (TCP/IP, HTTP, WebSockets, and others)
<b>Data Modeling</b>	Basic data exchange without hierarchical or relational models	Advanced data modeling with support for complex structures
<b>Scalability</b>	Limited to local networks; challenges in distributed systems	Scalable from small embedded devices to global cloud systems
<b>Security</b>	Basic security; prone to vulnerabilities	Built-in encryption, authentication, and user access control
<b>Interoperability</b>	Restricted to Windows; limited cross-platform compatibility	High interoperability across devices, platforms, and systems
<b>Standards Compliance</b>	Focused on individual specifications (e.g., DA, AE, HDA)	Unified standard combining multiple functionalities
<b>Extensibility</b>	Difficult to extend or adapt to new technologies	Flexible and extensible for future technologies
<b>Supported Environments</b>	Primarily industrial automation systems (local networks)	Suitable for Industry 4.0, IIoT, cloud, and edge computing
<b>Development Complexity</b>	Requires expertise in COM/D-COM	Modern APIs and SDKs simplify development
<b>Real-Time Performance</b>	Limited, especially over distributed networks	Optimized for high-performance real-time communication
<b>Birth Date</b>	Introduced in 1996	Released in 2006
<b>Supporters</b>	Mainly supported by Windows-based automation vendors	Widely supported by global automation and IT leaders

Table 13: Comparison between OPC Classic and OPC UA

OPC UA (Open Platform Communications Unified Architecture) is an open, secure, and versatile service-oriented architecture developed to facilitate interoperability between devices and systems in industrial automation. Introduced in 2008 as an evolution of the original OPC Classic, OPC UA

extends the capabilities of its predecessor by offering platform independence, enhanced security, and a richer information model. Unlike its predecessor, OPC UA is based on a service-oriented architecture (SOA), enabling it to operate across diverse environments, including embedded systems, cloud platforms, and traditional IT infrastructures. It supports various communication protocols, such as TCP/IP and HTTP, making it highly adaptable to modern industrial needs. In fact, unlike OPC Classic, which is primarily COM/DCOM-based and platform-specific, OPC UA is designed to work across various operating systems and supports communication over Ethernet without relying on Microsoft technologies (see Table 13 for all the differences).

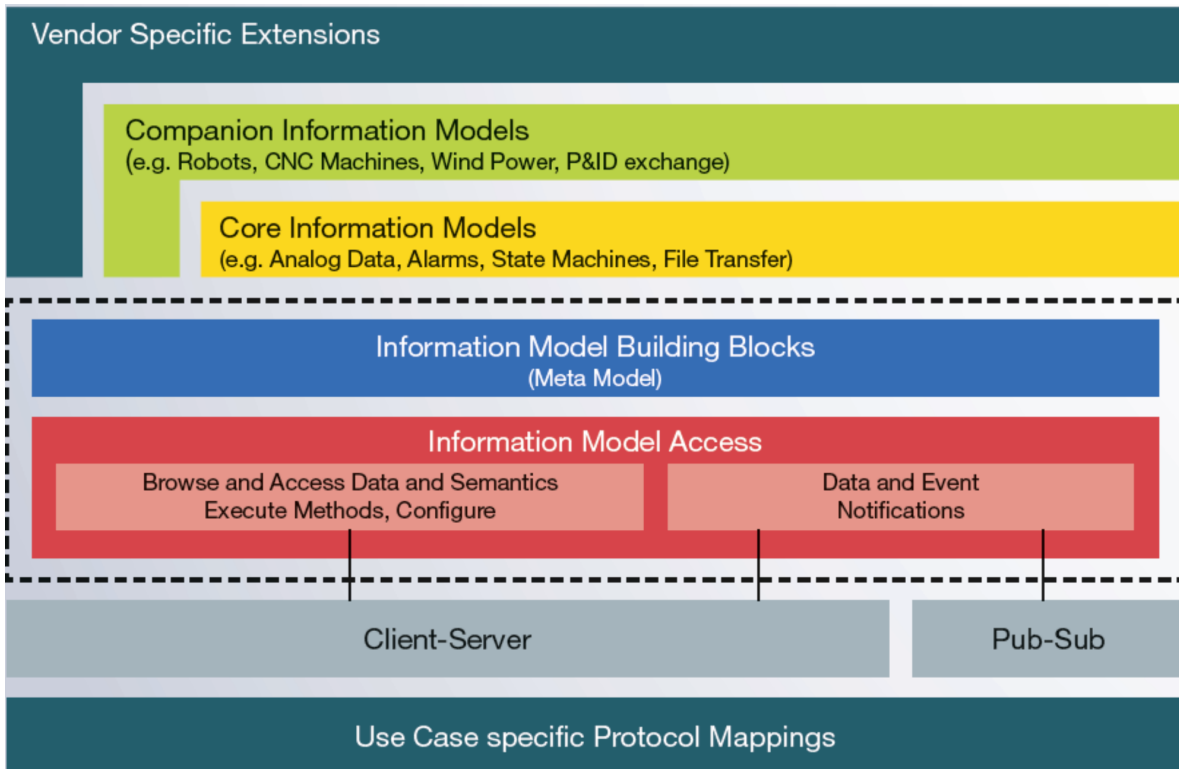


Figure 21: Multi-layered architecture of OPC UA

The OPC UA multi-layer architecture, as depicted in Figure 21, can be divided into three main blocks. Each layer in the figure represents a crucial aspect of OPC UA’s capabilities, enabling consistent, efficient, and secure data handling.

### 1. Network and Communication Layer

At the foundation of the architecture lies the communication infrastructure, which ensures seamless and reliable data exchange across systems:

- **Protocol Mappings:** This layer enables OPC UA to operate over different transport protocols, such as TCP and UDP. It supports multiple use cases, providing both high-performance and compatibility options.
- **Communication Paradigms:**
  - *Client-Server:* This traditional request-response model allows clients to interact with servers by reading and writing data or invoking methods. It is well-suited for scenarios where

explicit interactions are required.

- *Publish-Subscribe (Pub-Sub)*: This modern paradigm supports efficient, event-driven communication by allowing data producers (publishers) to send updates to multiple consumers (subscribers) simultaneously. It is ideal for large-scale, distributed systems with real-time requirements.

This foundational layer provides the transport and interaction mechanisms that support higher-level functionality.

## 2. Information Model Building Blocks

This block focuses on defining and providing the tools for interacting with data within OPC UA.

- **Information Model Access**: Users can browse and access structured data, execute methods, and configure systems. Additionally, this layer enables handling dynamic aspects like event notifications and real-time updates.
- **Meta Model (Building Blocks)**: At the core of the architecture lies the meta-model, which defines the fundamental constructs for creating information models. It provides the abstraction needed to represent complex systems, offering flexibility and consistency in data organization.

These building blocks make OPC UA a powerful framework for organizing, accessing, and interacting with structured information.

## 3. Custom Data Models and Extensions

The upper layers of the architecture enable the creation and extension of data models tailored to specific applications:

- **Core Information Models**: These predefined models provide the basic types and structures needed to represent fundamental concepts, such as analog data, alarms, state machines, and file transfer. They serve as the foundation for building more complex models.
- **Companion Specifications**: Designed for specific industries or domains (e.g., robots, CNC machines, or wind power), companion models extend the core information models with domain-specific semantics and structures.
- **Vendor-Specific Extensions**: At the highest level, vendors can customize models further to include proprietary features while ensuring interoperability within the OPC UA framework.

This block highlights the flexibility of OPC UA, allowing the development of custom data models while leveraging standardized building blocks and domain-specific extensions.

This modular design of OPC UA aligns perfectly with the needs of the proposed framework, enabling interoperability, secure communication, and efficient data management across the heterogeneous systems involved. Building on this foundation, the server will be developed by leveraging OPC UA's capabilities. In this context, the server will not only utilize OPC UA's core functionalities but also extend them by implementing a Vendor-Specific Information Model. This custom model will conform to the data structure and semantics defined by the framework proposed in this thesis, ensuring that the server aligns perfectly with the framework's requirements. By doing so, the server will act as the backbone of the system, enabling the seamless exchange of static and dynamic data while maintaining a high degree of flexibility and scalability.

### 4.3 Implementation

Once the system design procedure proposed by the framework is completed, all system components are well-defined.

To summarize:

- The mechatronic system is a Cartesian robot, controlled using Simulink Desktop RealTime. All system information has been represented using the proposed information model.
- The AR application will be developed using the Magic Leap2 headset, using the OpenXR specification, in Unity, making the application headset-independent. It has been decided that the mechatronic system will be tracked in a marker-less way, using its 3D model. To implement tracking, Vuforia will be used, alongside a review of the state of the art, highlighting the potential of this tool.
- Finally, the server will be an OPC UA server, where a custom information model will be implemented to represent the mechatronic system.

Figure 22 shows the components just listed.

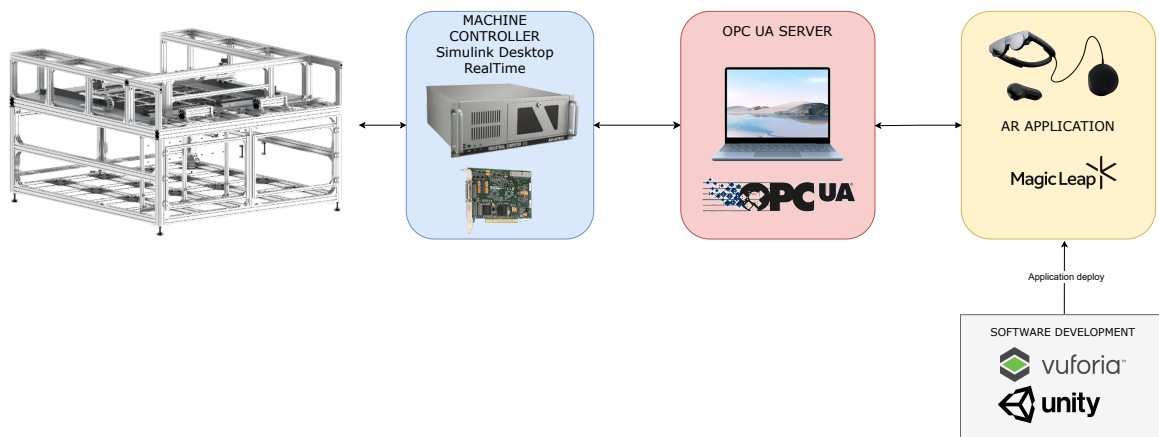


Figure 22: Architecture representation

At this point, the following sections will describe the steps to actually implement the defined design choices. It is worth noting that, although the design procedure ultimately defines the server due to its intermediary nature, in the implementation phase, the process moves in the opposite direction. The server is, in fact, the first to be developed, in order to easily test data exchange with the AR application and the mechatronic system.

### 4.3.1 Server

To create the OPC UA server, a Windows PC and the .NET libraries provided by the OPC Foundation were used. Several key files are needed, including:

- **Application Configuration:** This file is essential for properly configuring the OPC UA server. It typically contains settings such as the application name, endpoint type, and other configuration details related to security (e.g., certificates for encryption) and server behavior. This file is usually defined through C# code during server initialization, but an external `.config` file to manage some configurations dynamically can also be used.
- **Certificate files:** If the server uses security (encryption, authentication), you will need X.509 certificates for both the server and the clients. These certificates are generated and managed by the server, and their use is configured in the `ApplicationConfiguration`.
- **Nodeset2.xml (Information model file):** The `Nodeset2.xml` file contains the definition of the information model for the server, describing the nodes, data types, and relationships that the server will expose to users and OPC UA clients. The server loads this file to expose the data and structures defined by the information model.

The developed server must implement the information model defined by the framework, with which all the information of the mechatronic system is represented. To this end, it is necessary to have a custom `Nodeset2.xml` file to ensure that the model is correctly exposed by the server. There are three primary methods to create these models: graphically using specialized software, manually writing the `Nodeset2.xml` file, or using tools like the OPC Foundation's Model Compiler. Each method offers distinct advantages and limitations.

Creating the information model graphically with software tools allows developers to design models using an intuitive visual interface. Tools like graphical modelers often simplify the process by abstracting the underlying XML structure, making it more accessible for developers who are less familiar with the technical details of OPC UA. However, these tools can sometimes be less flexible, especially for complex or highly customized applications, and may not expose all the intricacies of the OPC UA specification.

Writing the `Nodeset2.xml` file manually offers complete control over the model's structure and content. This approach is particularly useful for developers who require precise alignment with specific requirements or who need to incorporate custom features. On the downside, manual editing is prone to human error and can be time-consuming, especially for large or complex models. It also demands a deep understanding of the OPC UA standard and its XML schema, which might pose a challenge for less experienced developers.

The third method, using the Model Compiler, automates much of the process by taking a higher-level input file conforming to the `UA Model Design.xsd` schema and generating the `Nodeset2.xml` file. This approach balances automation with flexibility. The Model Compiler significantly reduces the likelihood of errors and ensures compliance with OPC UA standards. However, setting up and using the tool requires familiarity with its specific syntax and workflows, which may involve an initial learning curve.

Given the importance of information model generation in OPC UA-based systems, significant research efforts are dedicated to simplifying and streamlining the process. Automating this phase not only reduces the risk of errors but also accelerates development and ensures consistency across implementations. For instance, studies like Automated generation of OPC UA information models — A review and outlook [95] emphasize the need for tools and methodologies to make model creation more efficient. These efforts are justified by the fact that the quality of the information model directly impacts the system’s interoperability, maintainability, and performance.

Other research contributions such as A Systematic Approach to OPC UA Information Model Design [96] highlight structured and automated methods for generating OPC UA information models, emphasizing the need for tools that simplify the development process while managing the protocol’s complexity. Similarly, Generation of OPC UA companion specification with eclipse modeling framework [97] and Evaluating domain-specific languages for the development of OPC UA based applications [98] explore innovative approaches to streamline model creation. Other studies, such as [99], [100], [101], and [102], investigate the use of UML diagrams or relational databases to bridge the gap between high-level conceptual designs and technical implementations, reinforcing the significance of research in this area to enhance accessibility and efficiency.

In this thesis, the Model Compiler is used to generate Nodeset2.xml files starting from an input file conforming to the UA Model Design.xsd schema. This approach ensures that the resulting information model is robust and adheres to OPC UA standards while allowing for structured and comprehensible design. By leveraging the capabilities of the Model Compiler, the process remains efficient and precise, demonstrating the critical role of tool-supported generation in creating high-quality OPC UA information models.

In OPC UA, the concept of nodes and relationships serves as the foundation for representing and organizing information. A node is a fundamental building block that encapsulates data or functionality, similar to an object in object-oriented programming. Nodes can represent various elements, such as physical devices, data points, or executable methods. Each node has attributes like a unique identifier (`NodeId`), a human-readable name (`BrowseName`), and a type that defines its properties and behavior (`ObjectType`, `VariableType`, etc.). Relationships between nodes, known as references, define how nodes are interconnected, forming a structured and navigable address space. These references, similar to associations or links in object-oriented design, can describe hierarchical relationships (e.g., `HasComponent`, `Organizes`) or more complex interactions (e.g., `HasProperty`, `References`).

Starting from the concepts of nodes and relationships in OPC UA, the proposed information model has been implemented. As a foundational step, all the required `ObjectTypes` were defined to serve as building blocks for the information model. These `ObjectTypes` provide a reusable and extensible basis for representing any mechatronic system and ensure that the model adheres to the standardized structure.

In Figure 23, the generic model is shown, including all the building blocks created as the foundation for the information model. In Figure 24, the UML notation proposed by OPC UA to represent information models is clarified.

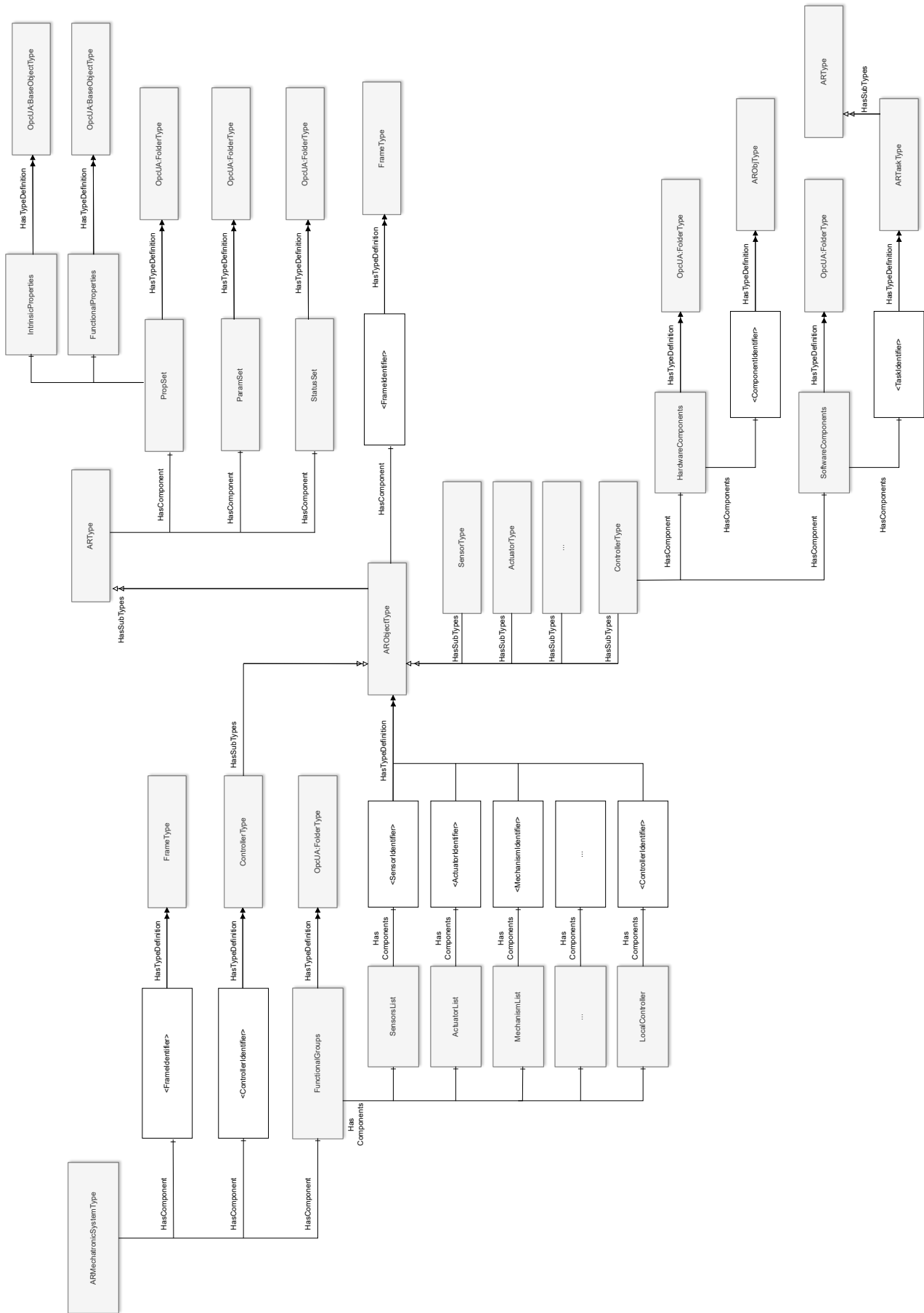


Figure 23: OPC UA information model

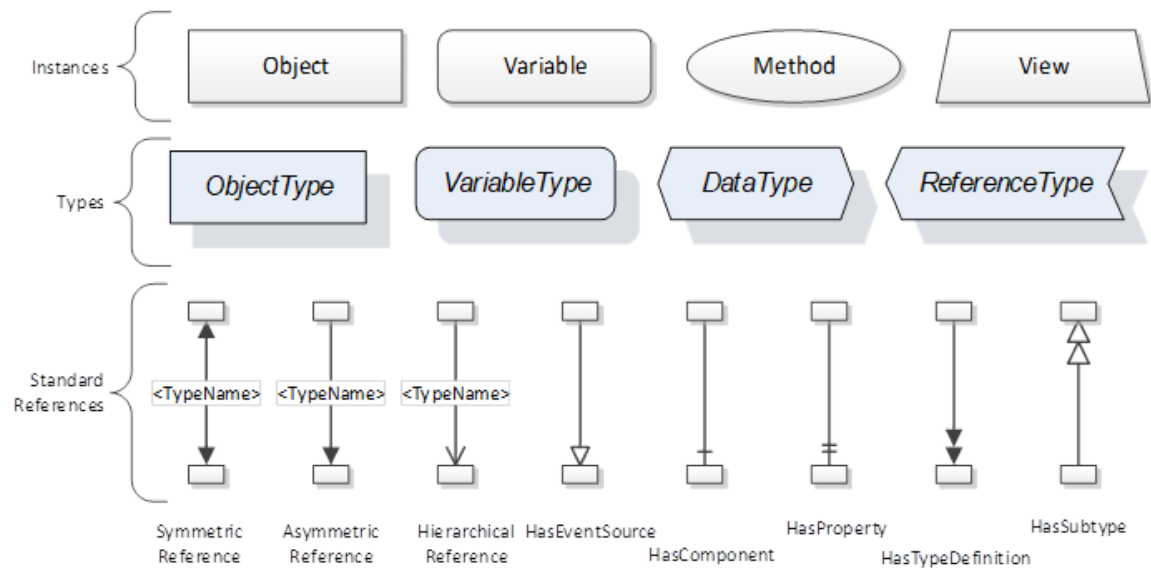


Figure 24: OPC UA UML Specification

Looking at the diagram, it can be observed that the graph structure previously presented has been maintained. The base node, ARMechatronicSystemType, as defined by the framework, contains the origin coordinate system (an object defined as a FrameType), the central controller of the system, and the functional groups, gathered into an object belonging to the OPC UA building blocks, namely a FolderType.

An ObjectType named ARTask has also been defined, which contains the three groups of information extensively discussed earlier: StatusSet, PropSet, and ParamSet. These three groups are conceptualized as FolderType nodes and will contain within them VariableType nodes corresponding to the data type being represented. For instance, the ActualValue of a proximity sensor will be a variable of type Boolean, while the ActualValue of an encoder will be of type Float.

From the ARType object, two subtypes were derived: ARObject, which add position information, and ARTask. From the ARObject, a series of objects have been derived to represent common components in mechatronic systems, such as sensors, actuators, etc. Starting from these building blocks, it is possible to describe the mechatronic system in question, namely the Cartesian robot, by incorporating all the information detailed in the Mechatronic System Information Modelling Phase section (3.3.2).

Once the required ObjectTypes were defined, they were used to implement the representation of the Cartesian robot. As previously discussed, both operations — defining the ObjectTypes and subsequently creating the Objects to build the mechatronic system model (which represent instances of the defined types) — were performed by writing an XML file compliant with the UA Model Design.xsd schema.

This file, by definition, can be compiled using the Model Compiler, also provided by the OPC Foundation, to generate the NodeSet2 file required by the OPC UA server.

```

<Object SymbolicName="MDLAB:ARMechatronicSystem"
  TypeDefinition="MDLAB:ARMechatronicSystemType">
  <Description>Cartesian Robot Base node</Description><Children>
  <Object SymbolicName="MDLAB:FunctionalGroups"
    TypeDefinition="OpcUa:FolderType"/>
  <Object SymbolicName="MDLAB:BaseFrame" TypeDefinition="MDLAB:FrameType"/>
  <Object SymbolicName="MDLAB:CentralizedController"
    TypeDefinition="MDLAB:ControllerType">
  <Children>
    <Object SymbolicName="MDLAB:SoftwareComponents" TypeDefinition="OpcUa:
      FolderType"/>
    <Object SymbolicName="MDLAB:HardwareComponents" TypeDefinition="OpcUa:
      FolderType"/>
    <Object SymbolicName="MDLAB:Frame" TypeDefinition="MDLAB:FrameType">
    <Children>
      <Variable SymbolicName="MDLAB:x" TypeDefinition="OpcUa:
        BaseDataVariableType"
        DataType="OpcUa:Float" AccessLevel="Read">
        <DefaultValue>
          <uax:Float>-1.785</uax:Float>
        </DefaultValue>
      </Variable>
      <Variable SymbolicName="MDLAB:y" TypeDefinition="OpcUa:
        BaseDataVariableType"
        DataType="OpcUa:Float" AccessLevel="Read">
        <DefaultValue>
          <uax:Float>0.905</uax:Float>
        </DefaultValue>
      </Variable>
      <Variable SymbolicName="MDLAB:z" TypeDefinition="OpcUa:
        BaseDataVariableType"
        DataType="OpcUa:Float" AccessLevel="Read">
        <DefaultValue>
          <uax:Float>2.258</uax:Float>
        </DefaultValue>
      </Variable>
      <Variable SymbolicName="MDLAB:Fixed" TypeDefinition="OpcUa:
        BaseDataVariableType"
        DataType="OpcUa:Boolean" AccessLevel="Read">
        <DefaultValue>
          <uax:Boolean>1</uax:Boolean>
        </DefaultValue>
      </Variable>
      <Variable SymbolicName="MDLAB:Initialized" TypeDefinition="OpcUa:
        BaseDataVariableType"
        DataType="OpcUa:Boolean" AccessLevel="Read">
        <DefaultValue>
          <uax:Boolean>1</uax:Boolean>
        </DefaultValue>
      </Variable>
    </Children>
  </Object>
  </Children>
</Object>
</Children>
</Object>

```

Listing 1: Implementation of ARMechatronicSystem using UA Model Design.xsd schema

An excerpt of the written XML file is shown for illustrative purposes in 1. The code demonstrates the definition of the root node of the information graph. As can be observed, an OPC UA object is defined with *TypeDefinition = "MDLAB : ARMechatronicSystemType"*, which corresponds to the created *ObjectType*. It is noteworthy that all the building blocks developed are part of a namespace named MDLAB, representing the Mechatronic and Mechanical Dynamic Lab where the project was carried out. The MDLAB namespace groups all custom elements, aligning with OPC UA's recommendation for extensibility and namespace management.

The excerpt of the written XML file defines the nodes of the information model compliant with the OPC UA framework. Some aspects that characterise the way these files are drafted are:

- Definition of a node: the root node is an object defined with:
  - SymbolicName="MDLAB:ARMechatronicSystem": A symbolic name representing the mechatronic system.
  - TypeDefinition="MDLAB:ARMechatronicSystemType": Specifies the type of the root node, based on the ARMechatronicSystemType ObjectType defined in the MDLAB namespace.
  - *¡Description¿*: A textual description of the node, useful for understanding its purpose (e.g., "Cartesian Robot Base node").
- Node Relationships: The root node's hierarchy is structured using the *¡Children¿* element, which represents child nodes, and hasChild hierarchical relation. In this case, the root node contains:
  - FunctionalGroups: A node of type FolderType for organizing functional groups.
  - BaseFrame: A node of type FrameType representing the base coordinate system.
  - CentralizedController: A node of type ControllerType representing the system's central controller.
- Internal Node Definitions:
 

Each child node can also have its own children. For instance, within CentralizedController:

  - SoftwareComponents: A FolderType node organizing software components, i.e. Tasks.
  - HardwareComponents: A FolderType node organizing hardware components, i.e. Ni acquisition boards.
- Defining Variables:
 

Variables are nodes that encapsulate data. They are defined within *¡Children¿* elements of objects like Frame. For example: Position Variables (x, y, z):

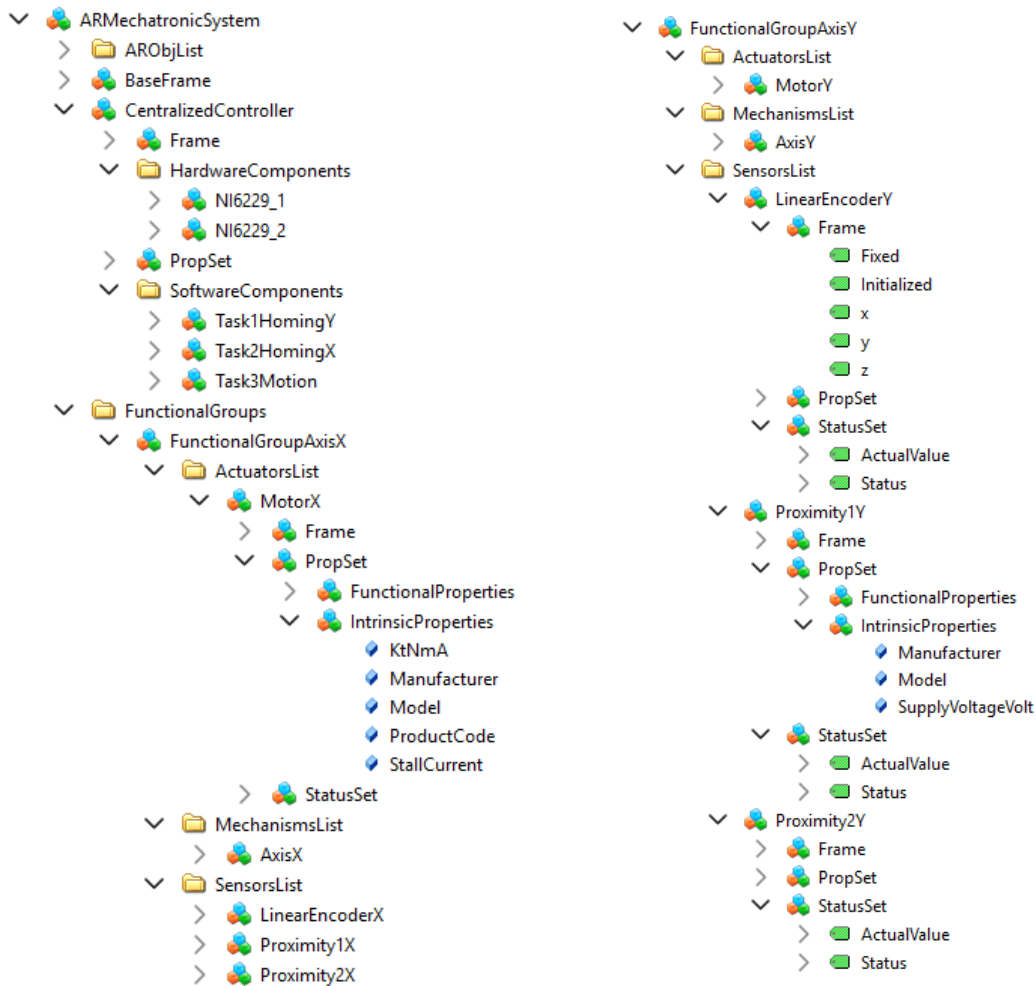
  - SymbolicName="MDLAB:x": A symbolic name for the variable.
  - TypeDefinition="OpcUa:BaseDataVariableType": Specifies the variable type.
  - DataType="OpcUa:Float": Indicates that the variable holds a float value.
  - AccessLevel="Read": Marks the variable as read-only.
  - Default values are defined using the *¡DefaultValue¿* element, ensuring initial state consistency.

The OPC UA server created with the .NET library exposes the following endpoint:

This endpoint allows clients to connect to the server and interact with the data it exposes. In the string, it is possible to see the protocol used, i.e. tcp, defined as compatible with both the mechatronic system controller and the AR application, during the design phase.

The information model exposed by the server, which represents all the information related to the Cartesian robot according to the model defined by the framework, can be validated using the UAExpert client. As shown in Figure 25, UAExpert can connect to the provided endpoint, and provide the structured view of the information model exposed by the server, displaying all variables and objects in their graph form.

Figure 25: UAExpert Adress space



### 4.3.2 Mechatronic system controller upgrade

The mechatronic system under examination, the Cartesian robot, is controlled using an industrial robot interfaced with Simulink Desktop Real-Time (SDRT). SDRT is a powerful tool that enables real-time simulation and testing of dynamic systems by executing Simulink models directly on a dedicated real-time kernel. To extend the robot’s capabilities and enable seamless data exchange with the server, the system was connected via cable to a local network shared with the OPC UA server and the AR headset.

Simulink Desktop Real-Time supports OPC UA through the Industrial Communication Toolbox. This toolbox provides specialized blocks for integrating OPC UA communication into Simulink models. It allows for the configuration of OPC UA clients and servers, facilitating secure and reliable data exchange in industrial automation systems. By leveraging these blocks, a subsystem was developed to cyclically write the values of key OPC UA nodes, such as sensor states and task progress, and read nodes containing user-defined parameters set via the AR interface.

This approach enabled an extension of the controller’s functionality without disrupting its core operation. Similarly, this method could be applied to systems using PLCs or other types of controllers, showcasing the versatility and adaptability of the proposed solution.

### 4.3.3 AR Application development

This section delves into the implementation of the monitoring and control AR application defined in the previous sections, developed for the Magic Leap 2 (ML2) platform.

Magic Leap 2 devices can be programmed in three primary ways, depending on the developer’s expertise and the desired application. The first method involves using Unity, a powerful game engine that, when combined with OpenXR, allows for the creation of immersive AR experiences while ensuring cross-device compatibility and streamlined deployment. The second approach is through Unreal Engine, which provides robust tools for high-fidelity rendering and access to Magic Leap’s features via its integrated plugins. Lastly, for those who prefer a lower-level programming interface, direct development using the Magic Leap SDK is available, enabling precise control over device capabilities, often preferred for applications requiring integration into existing native frameworks.

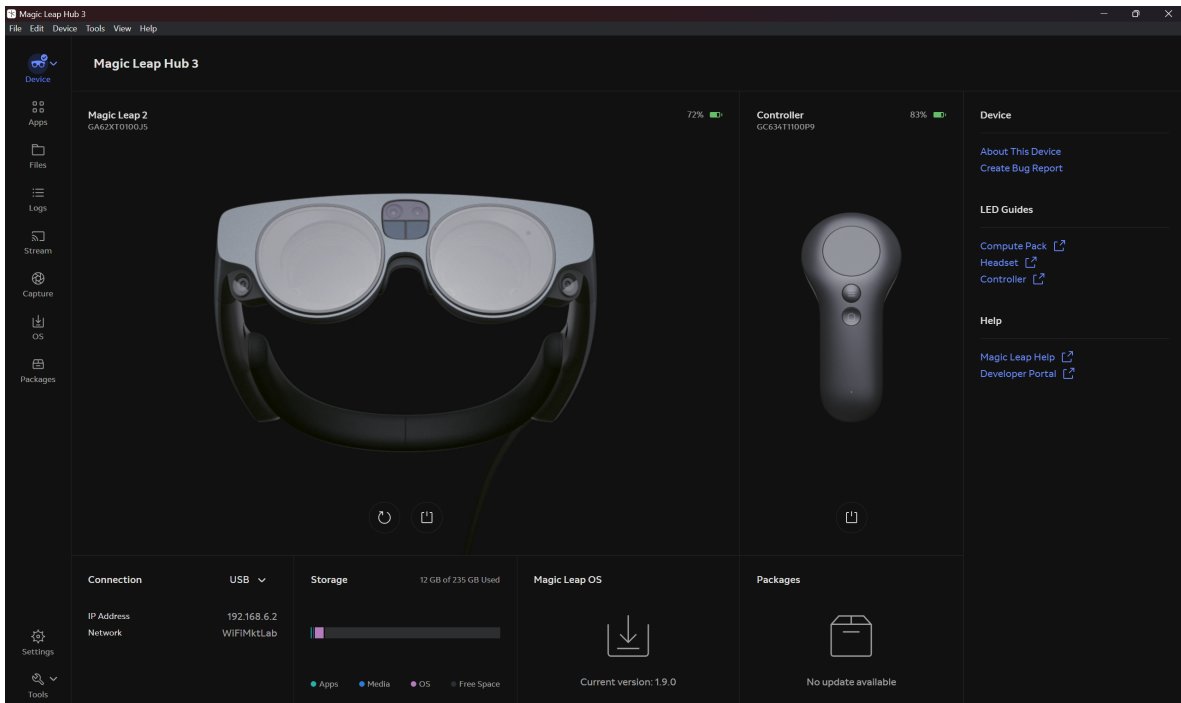


Figure 26: Magic Leap Hub 3

For this application, Unity was chosen specifically for its support of OpenXR, ensuring platform independence and broader compatibility, as requested in the design phase.

Supporting this development process is the Magic Leap Hub 3 [103], Figure 26, a comprehensive tool for managing Magic Leap2 devices, deploying applications, and accessing essential resources. The Hub streamlines tasks such as SDK installation, Android tool configuration, and device-to-environment connections, while also offering features like an emulator for testing without hardware and robust debugging tools, making it indispensable for efficient Magic Leap 2 development.

To clarify the development process, the AR application can be summarised in three main parts, as proposed by the framework in section AR Application Development Model: Workflow Based on the Information Model 3.4: the setup phase, the preparation of augmented objects, and the cyclic operation phase.

Unity is a game engine that allows developers to create interactive applications by combining 3D models, scripts, and user interface elements. At the core of Unity’s scripting system is the MonoBehaviour class, which serves as the foundation for creating behavior scripts. These scripts are written in C# and define operations performed by GameObjects at different points during the application’s lifecycle. Key methods in MonoBehaviour include:

- Awake: Called when the script instance is first loaded. This is often used for initialization tasks that need to occur before the application begins running.
- Start: Executed once, after all objects have been initialized and just before the first frame update. This is ideal for setup operations that rely on other initialized components.
- Update: Called once per frame, allowing developers to handle dynamic behaviors and interactions. This is particularly useful for continuous tasks, such as updating object states or handling user inputs.

In this application, a generic script serves as the controller, orchestrating the setup phase and initiating the application’s core processes.

## Setup Phase

The application begins by establishing a connection to the OPC UA server, a crucial step that lays the foundation for subsequent interactions. The general controller script takes responsibility for this connection, configuring the server endpoint and defining the security policies. Once a session is established, the application prepares for dynamic updates by subscribing to relevant data nodes within the OPC UA server. This connection is essential for accessing real-time information and ensuring seamless data flow.

Simultaneously, the system initializes tracking mechanisms using Vuforia. Vuforia’s capabilities are employed to locate the origin of the mechatronic system within the physical environment. Once identified, SLAM (Simultaneous localization and mapping) ensures continuous spatial alignment, providing the precision needed for accurately placing augmented markers. This tracking approach guarantees that the augmented reality elements remain properly aligned with the physical system, even as the user moves around. A separate section will be dedicated to the functionality of Vuforia 4.3.3.

## Augmented Object Preparation Phase

After establishing the connection and spatial tracking, the application transitions to preparing augmented objects. This involves exploring the OPC UA address space to identify objects of interest. The general controller script plays a pivotal role here, scanning the address space for nodes that conform to the ARObjct schema. These nodes represent components or elements of the mechatronic system.

For each identified ARObjct, the application retrieves its spatial coordinates relative to the system's origin (obtained by Vuforia). Using these coordinates, a marker is instantiated in Unity to represent the ARObjct in the augmented reality environment. These markers are not mere placeholders; they are GameObjects equipped with customizable meshes to visually differentiate their roles and functions. Each marker is also assigned a script that handles its behavior, including cyclic updates from the OPC UA server and user interactions.

Special consideration is given to Controller objects, which serve a unique role in the system. These objects are represented by enhanced markers and are equipped with a dedicated script that not only visualizes their data but also aggregates and displays information about related tasks. This additional layer of functionality allows users to interact with higher-level control elements of the mechatronic system.

## Cyclic Operation Phase

In the final phase, the application enters its operational mode, where augmented objects function autonomously. Each ARObjct operates independently, thanks to its associated script, which is responsible for both updating the object's data and managing user interactions.

These updates occur cyclically (Update method of Monobehavior class), with the script querying the OPC UA server to fetch the latest information about the object's state. Additionally, when a user modifies parameters through the interface, these changes are written back to the OPC UA server, ensuring that the system remains synchronized and the control actions are reflected in real time.

Above each marker, a dynamic canvas is instantiated to present the retrieved data. The canvas provides a user-friendly interface that displays static properties, real-time state information, and adjustable parameters. This interface is designed to be intuitive and customizable, enabling users to navigate and interact with the data effortlessly. For Controller objects, the canvas is further enhanced to include information about associated tasks, offering a comprehensive view of the system's operational hierarchy.

The application's modular design ensures that it can adapt to a wide variety of mechatronic systems. By adhering to the proposed data representation model, the system can be configured with appropriate graphical interfaces and augmented markers, tailoring the user experience to specific requirements.

Once the structure of the developed application is defined, a paragraph will be dedicated to tracking using Vuforia, followed by an in-depth discussion on input management with OpenXR.

## Tracking 3D models with Vuforia

In defining the requirements for the AR application, the goal was to develop a marker-less monitoring system. By leveraging the availability of the 3D model of the mechatronic system to be monitored, combined with a spatial description of its components, all the necessary inputs are provided to meet this objective. The object can be tracked using its model, allowing the system's origin coordinate position to be identified. This enables coherent positional information to be assigned within the system, ensuring accurate visualization and interaction with data in the augmented environment.

To implement 3D model tracking, Vuforia was chosen, a robust and well-established augmented reality platform developed by PTC (Parametric Technology Corporation). Widely used in industrial research and development, Vuforia provides powerful tools for tracking a broad range of elements in the physical world. It supports image targets, enabling the recognition of 2D images such as logos, illustrations, or photographs, traditional markers like QR codes or patterned tags, which remain effective even in challenging lighting conditions, and area targets, which allow for spatial mapping of environments using spatial scans.

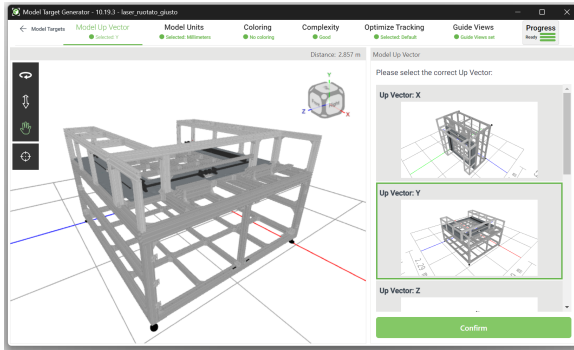
Among its most advanced features is the support for model targets, which enable the recognition and tracking of complex 3D objects based on CAD files or 3D representations. This capability is particularly suited to industrial contexts, where machines or mechanical components need to be seamlessly integrated into augmented environments. Vuforia's ability to track multi-targets, combining multiple surfaces or edges into cohesive reference systems, further enhances its flexibility.

This versatility makes Vuforia the ideal choice for this project, where the focus is on monitoring a mechatronic system using 3D models as a primary tracking method. By leveraging this advanced functionality, we ensured the application meets the requirements for accurate and reliable object recognition in an industrial setting.

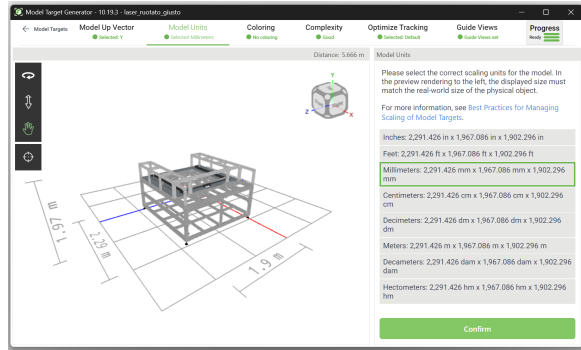
The choice of Vuforia reflects the focus of this project, which is not on developing novel tracking algorithms but rather on creating a framework that facilitates the integration of AR applications with mechatronic systems. Vuforia's proven reliability and widespread adoption make it an ideal choice for achieving this objective, ensuring a streamlined and efficient development process.

Once Vuforia was selected, the next step involved demonstrating how to generate and utilize a model target dataset within Unity, the game engine used to develop the application for the Magic Leap 2 headset. The model target creation process begins with importing the 3D model into Vuforia's Model Target Generator. Here, several critical steps are performed to optimize the model for AR tracking, summarized in Figure 27 and described below.

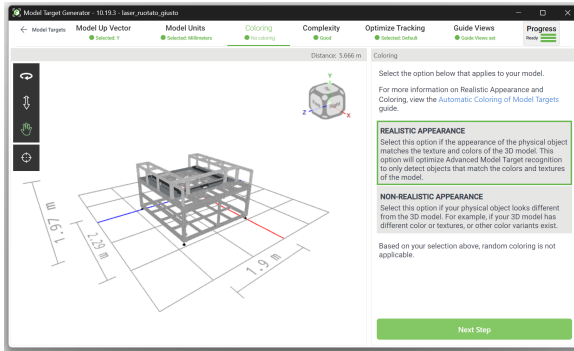
- **Model Up Vector:** Defining the orientation of the model by specifying the up vector is a critical step in ensuring proper alignment of the target in space during tracking. The up vector determines the direction in which the model is considered "upright" relative to the tracking environment.
- **Model Units:** The scale of the 3D model must be accurately defined to maintain consistency between the virtual representation and the corresponding physical object. This involves specifying the correct unit of measurement (e.g., millimeters, centimeters, or meters) used to create



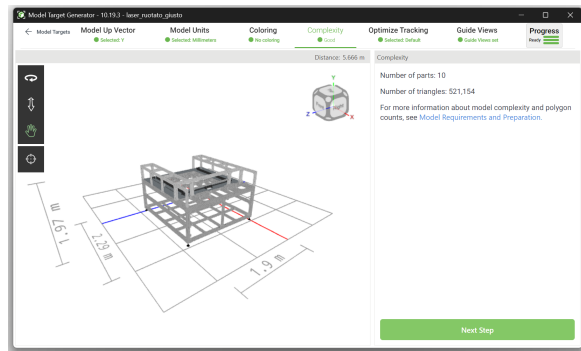
(a) Model Up Vector



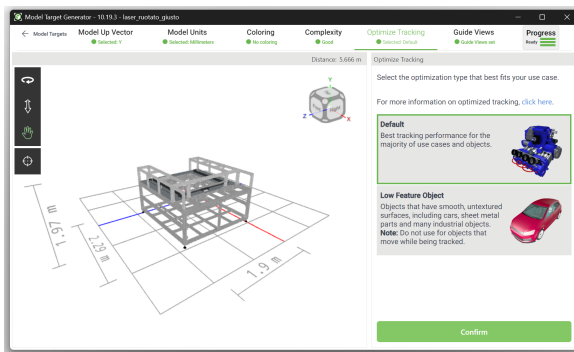
(b) Model Units



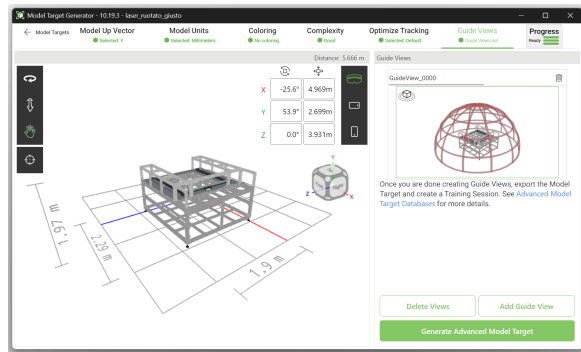
(c) Coloring



(d) Complexity



(e) Optimize Tracking



(f) Guide Views

Figure 27: Model target generator overview

the model.

- **Coloring:** Adjusting the model's appearance by enhancing its color or texture improves the recognition process by highlighting key features that the tracking algorithm relies on. High-contrast areas or distinctive features of the model can be emphasized to make it easier for the system to identify and lock onto the target. This step is particularly useful for models with uniform surfaces, where additional detail helps the algorithm achieve better accuracy.
- **Complexity Optimization:** Simplifying the geometry of the 3D model where necessary ensures efficient tracking performance without sacrificing essential details. This process involves reducing the polygon count or removing unnecessary elements from the model that do not contribute to the tracking algorithm. By striking a balance between detail and simplicity, the computational load is minimized while retaining the model's critical features for accurate tracking.
- **Tracking Optimization:** Configuring specific parameters to enhance tracking reliability under various environmental conditions is an essential step. This includes adjusting sensitivity settings to account for different lighting conditions, motion speeds, or distances between the camera and the target. Fine-tuning these parameters ensures that the tracking remains stable and accurate even in challenging scenarios, such as in dynamic industrial environments.

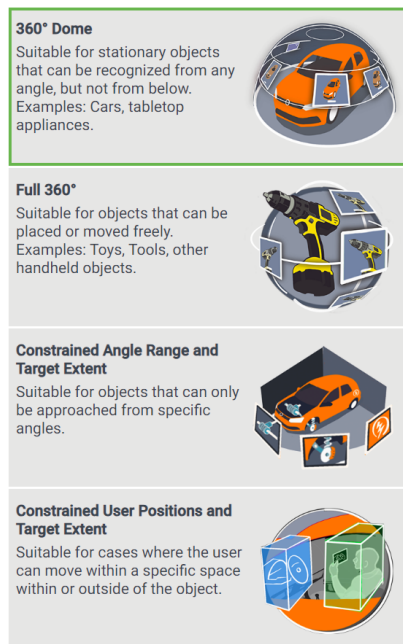


Figure 28: Guide views

- **Guide Views:** The guide views define the expected alignment between the 3D model and the real object during the tracking initialization process. These configurations are critical to guide the tracking algorithm in recognizing and engaging the model from different viewpoints. There are two types of guide views, standard and advanced. In the case of standard guide views, a well-defined position is defined with which the user should approach the target. For this purpose, the application displays visual aids to help the user align the camera with the model. A more sophisticated approach is required for advanced model targets on the other hand. For advanced targets, sets of positions are considered that the user can assume to initiate tracking.

The system performs a more complex tracking process that involves training the model with multiple views. This training process teaches the system to recognize the object from different perspectives, allowing it to reliably track the model without requiring the user to align the camera in a specific way. This flexibility allows the system to detect the model from different angles and orientations, making advanced model targets more suitable for the user's needs.

The available advanced modes are shown in Figure 28.

For the cartesian robot investigated in this study, a 360°Dome-shaped advanced guide view was employed, as this configuration best accommodates the limited viewing angles imposed by the robot's physical constraints. Dome-shaped guide views provide a comprehensive visualization of the target, ensuring consistent tracking from the constrained viewpoints typical of this setup. The representation of the dome of views from which tracking can begin is shown in the Figure 29.

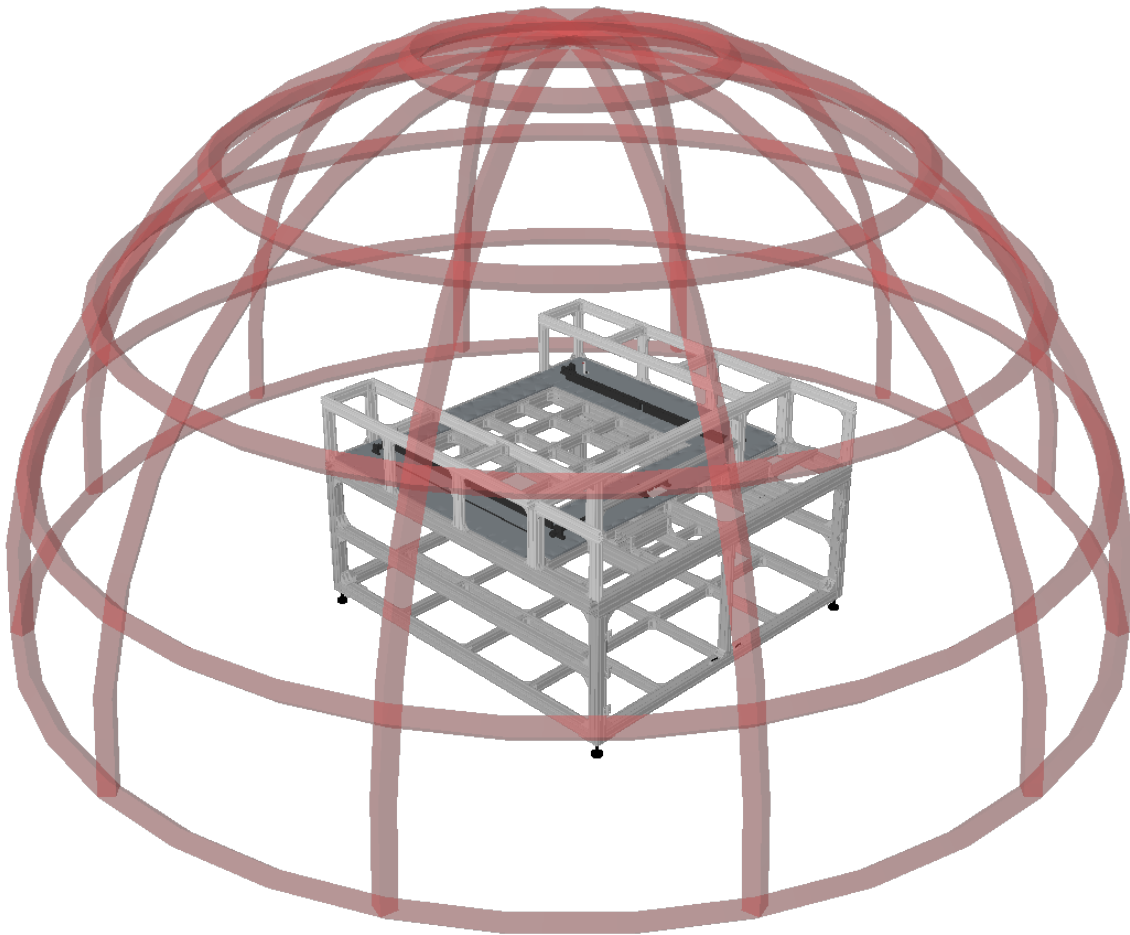


Figure 29: Representation of perspectives for tracking the Cartesian robot

Each of these steps was carefully executed, and the corresponding settings were refined to achieve optimal performance in the final application. By following this structured approach, the generated model target was seamlessly integrated into Unity, enabling efficient development of the AR application. The final result is a robust system capable of accurate and reliable monitoring of the cartesian robot.

## User Inputs: Action maps

Once the various augmented markers are instantiated in the correct positions of the system to be monitored, the user can interact with them through graphical interfaces that display data such as status and properties, and modify parameter values. To interact with the application, the user has access to a controller, which allows them to select and interact with objects using a virtual ray. The controller also has various buttons, each of which can be assigned specific functions.

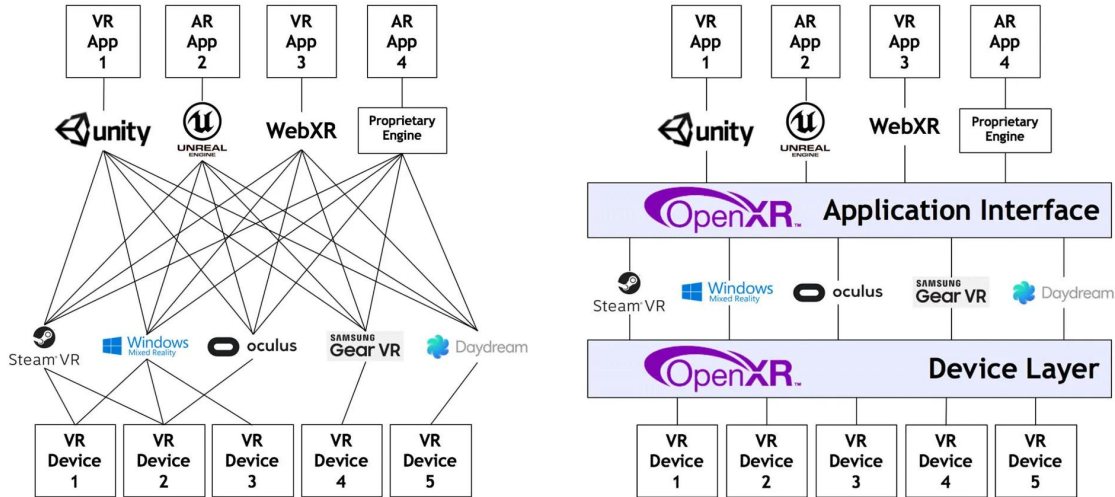


Figure 30: OpenXR Structure

In Unity, user input management is handled through the Input System (the evolution of the previous Input Manager), by creating Action Maps. Action Maps allow you to define a series of actions, which are then called within scripts and bound to specific inputs. In this way, for example, an action can be defined and linked to different bindings based on the device connected to the application (for instance, the Click() action could be triggered by a mouse click, a controller button, or a hand gesture, etc.). This method of handling input makes it possible to create more generic applications, abstracting the physical input from the action it triggers.

Considering the context of headsets, an application could be developed to be compatible with various headsets, connecting actions to specific bindings for each device. However, this operation would become complex, as it would require in-depth knowledge of each system's specific SDKs. To overcome this challenge, the OpenXR specification was created, which, as previously introduced, simplifies the creation of cross-platform applications.

OpenXR simplifies the development of XR applications by eliminating the complexity of directly interfacing with multiple device-specific SDKs. As illustrated in Figure 30, before OpenXR, developers had to create separate implementations for each device and runtime, leading to a web of dependencies and incompatibilities. OpenXR consolidates this into a single, unified Application Interface, allowing XR applications to interact seamlessly with various hardware through a standardized Device Layer. This abstraction significantly reduces development overhead and enhances portability between platforms.

The following Figures 31 and 32 showcases action maps for Magic Leap 2's based on OpenXR. As seen in the two images, the panel is composed of three sections. On the left, all the action maps are displayed, which in this specific case represent the components that can trigger interactions: the headset, the controller, the eyes, and the hands. By selecting a specific action map, you can see all the actions it contains, below which the defined bindings are listed. Depending on what is selected (actions or bindings), the properties of the selected item can be viewed and modified in the third section.

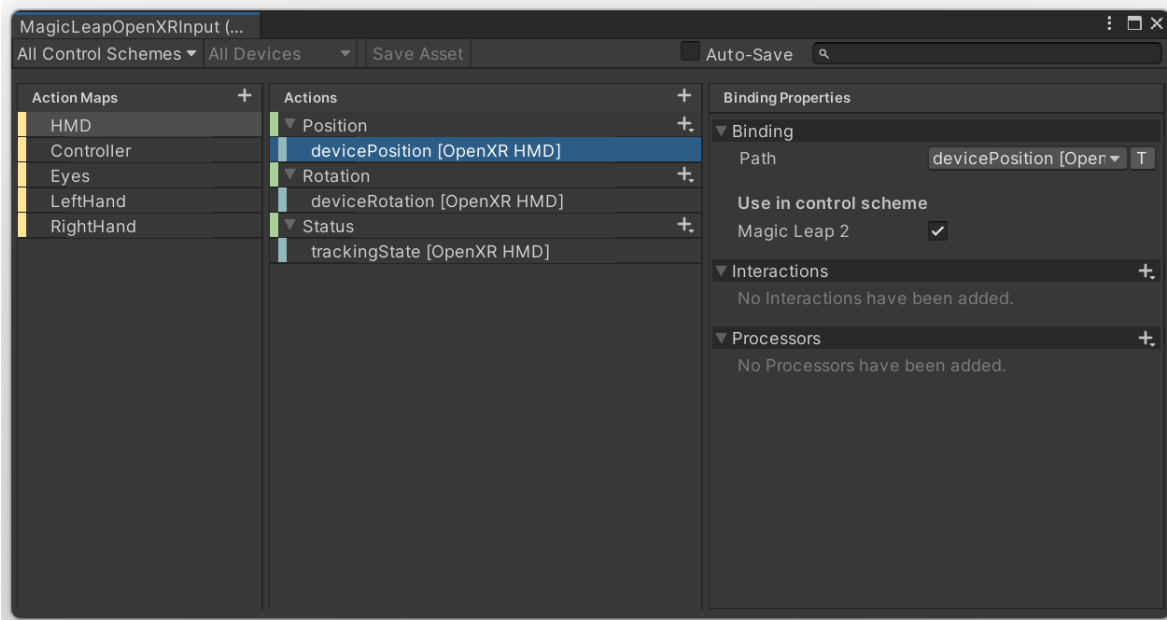


Figure 31: Action Map HMD

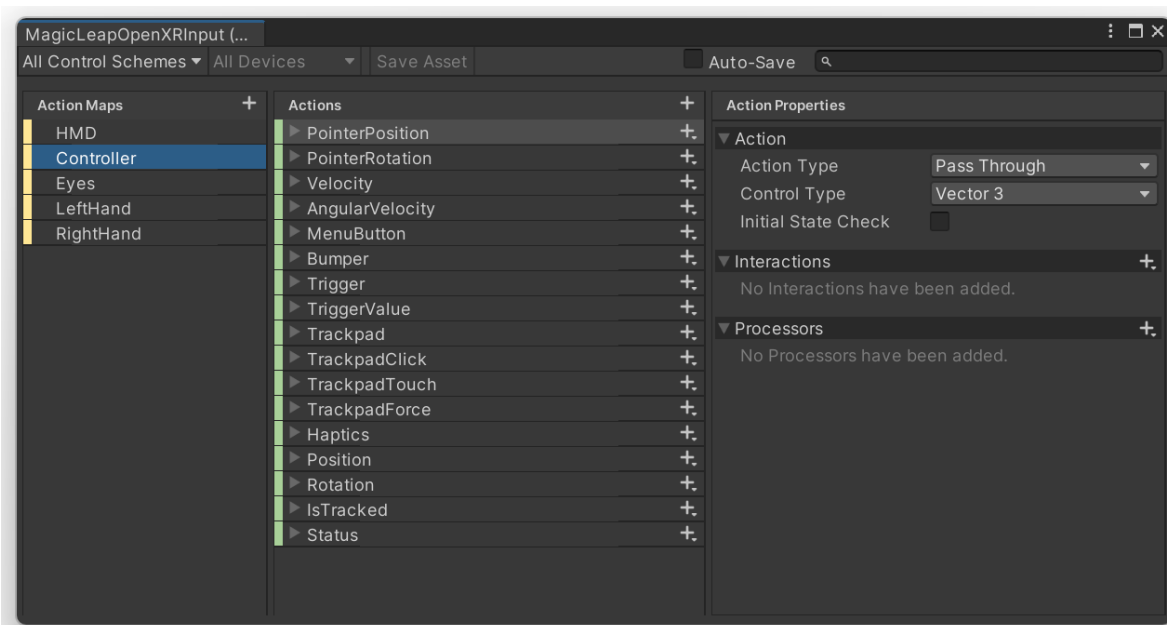


Figure 32: Action Map Controller

A summary of all the actions available from the OpenXR standard is provided in Table 14. Specifically, the table also shows the return type of each action, which can be used in the program's scripts.

In the developed application, the inputs from the headset were used for tracking and localization, while the controller inputs were used for interaction with the canvases. By developing the application in this way, any headset that supports the standard can be used without needing any modifications, thus meeting the defined compatibility requirement.

Table 14: OpenXR Action maps details

<b>Action Map</b>	<b>Action Name</b>	<b>Return Type</b>	<b>Description</b>
HMD	Position	Vector3	Position of the HMD in world space.
HMD	Rotation	Quaternion	Orientation of the HMD in world space.
HMD	Status	Integer	Tracking status of the HMD.
Controller	PointerPosition	Vector3	Position of the controller pointer in world space.
Controller	PointerRotation	Quaternion	Orientation of the controller pointer.
Controller	Velocity	Vector3	Linear velocity of the controller.
Controller	AngularVelocity	Vector3	Angular velocity of the controller.
Controller	MenuButton	Button	Indicates if the menu button is pressed.
Controller	Bumper	Button	Indicates if the bumper button is pressed.
Controller	Trigger	Button	Indicates if the trigger is pressed.
Controller	TriggerValue	Axis	Pressure value of the trigger.
Controller	Trackpad	Vector2	2D input from the trackpad.
Controller	TrackpadClick	Button	Indicates if the trackpad is clicked.
Controller	TrackpadTouch	Button	Indicates if the trackpad is touched.
Controller	TrackpadForce	Axis	Force applied to the trackpad.
Controller	Haptics	Haptic	Provides haptic feedback.
Controller	Position	Vector3	Position of the controller in world space.
Controller	Rotation	Quaternion	Orientation of the controller in world space.
Controller	IsTracked	Button	Indicates if the controller is being tracked.
Controller	Status	Integer	Tracking status of the controller.
Eyes	GazePosition	Vector3	Position of the user's gaze in world space.
Eyes	GazeRotation	Button	Orientation of the user's gaze.
Hands	IsTracked	Button	Indicates if the hand is being tracked.
Hands	TrackingState	Integer	Tracking state of the hand.

Action Map	Action Name	Return Type	Description
Hands	Position	Vector3	Position of the hand in world space.
Hands	Rotation	Quaternion	Orientation of the hand in world space.
Hands	Grip	Pose	Pose of the hand grip.
Hands	GraspValue	Axis	Value indicating grasp strength.
Hands	GraspReady	Button	Indicates readiness for a grasp gesture.
Hands	Aim	Pose	Pose for aiming.
Hands	AimValue	Axis	Value indicating aiming strength.
Hands	AimReady	Button	Indicates readiness for an aim gesture.
Hands	Pinch	Pose	Pose for a pinch gesture.
Hands	PinchValue	Axis	Value indicating pinch strength.
Hands	PinchReady	Button	Indicates readiness for a pinch gesture.
Hands	Poke	Pose	Pose for a poke gesture.
Hands	Palm	Pose	Pose of the palm.

#### 4.4 Conclusions

In this chapter, the steps defined by the framework, presented in Chapter 3, were followed to develop a system for real-time human-machine interaction through augmented reality. In particular, the procedure allowed the following to be defined:

- The requirements and, consequently, the specifications for the AR application, enabling the selection of the appropriate device and the software needs for implementation. Specifically, the device used is a see-through Magic Leap 2 headset. This choice satisfies all the requirements but is not strictly binding, as the OpenXR standard was used to develop an application compatible with all headsets supporting this standard.
- The information representation model was applied to the Cartesian robot, defining two functional groups corresponding to the two axes that make up the mechatronic system.
- Having defined the characteristics of the AR application and knowing the technical details of the pre-existing controller of the mechatronic system, it was possible to design the intermediary between the two systems. Specifically, both clients support the TCP protocol, so an OPC UA server was chosen as the intermediary to handle not only communication but also the representation of the proposed data model, leveraging the capabilities of OPC UA.

Once the design phase defined by the procedure was completed, the implementation phase began, starting with the implementation of the OPC UA server, followed by the extension of the existing Cartesian robot controller and the AR application. For the latter, the functional scheme proposed by the framework was followed to exploit the data model used, making it independent of the specific

mechatronic system and suitable for interacting with any system modeled according to the defined standard.

The result is a system that enables real-time interaction between the two components while decoupling their communication via the OPC UA server.

Discussing the complexity of the implemented system highlights how standardizing this process could facilitate the broader adoption of augmented reality (AR) in the industry.

Specifically, the development of the OPC UA server using the .NET libraries from the OPC Foundation is not particularly complex in terms of the server application's core functionality. However, the configuration files, including the information model representation and server configuration, are intricate XML files. In this case, the server's implementation and configuration alone (excluding external libraries) amount to 5,000 lines of code, 158 files, and 21 folders. Utilizing a predefined namespace could make much of this code reusable as a starting point, simplifying the development process.

Similarly, the development of the AR application was carried out using the Unity game engine, involving a total of 63,349 files and 4,002 folders (with 61,141 files and 3,677 folders attributed to external libraries) and over 3,500 lines of custom script for the application. Leveraging a game engine is inherently complex and required extensive study to proceed with the implementation. Creating a basic application integrated with a standardized data model could significantly reduce the time and expertise needed for such development.

Throughout the chapter, an effort was made to discuss the system implementation at a reduced complexity level, but it should be noted that, in practice, the system's implementation was a highly complex aspect of this thesis work.

In the next chapter, the results obtained will be presented and discussed.

## 5 Results

In the previous chapters, a framework was proposed to address challenges related to the use of augmented reality (AR) in industrial settings for the development of applications enabling effective human-machine interaction. Subsequently, the practical implementation of the proposed procedure was demonstrated using a case study centered on a Cartesian robot. Specifically, the application of the framework allows the integration of two systems—the AR application and the mechatronic system—to monitor the components by visualizing both their static information (intrinsic and functional properties) and dynamic data (real-time state), and to control the process by enabling the user to adjust system tasks through specific parameters.

This chapter presents and discusses the results achieved, focusing on the AR application developed to interface with the Cartesian robot via an OPC UA server. Before delving into this aspect, the simulated environment used during the application’s development will be described. Instead of directly developing the application on the Magic Leap 2 headset, the algorithms were initially implemented following the specified guidelines by using the 3D model of the robot and simulating user interactions with the virtual system. This preliminary step is detailed not only because it simplified the development process—allowing for greater control through local implementation—but also because it resulted in the creation of a digital twin of the object. This digital twin could potentially serve as a valuable tool for remote monitoring.

### 5.1 Development in Simulated Environment: Achieved Results and Foundations for a Digital Twin for Remote Monitoring

Before diving into the details of the simulation, it is important to introduce the Unity interface, as it will be referenced in the upcoming images. The Unity editor is divided into several key panels:

- **Scene View:** This panel allows to visually manipulate and design the elements in the 3D scene, such as objects, lights, and cameras.
- **Game View:** This panel provides a real-time preview of what the application will look like when running. It displays the output from the active camera in the scene.
- **Hierarchy Panel:** This lists all the objects currently present in the scene in a hierarchical structure, allowing for easy organization and selection.

These components work together to enable efficient design, debugging, and testing of Unity projects. In the following sections, screenshots of the Unity interface will illustrate how the simulation environment was developed and tested.

In Figure 33, the developed application is shown within the Unity interface, highlighting the panels described earlier. As mentioned in Section 3.4, \*AR Application Development Model: Workflow Based on the Information Model\*, the application consists of three main phases:

- **Setup:** This phase includes the connection to the OPC UA server and the localization of the object. In this simulation, only the server connection is implemented, while the localization will be integrated later when the application is deployed on the Magic Leap headset.

- **AR Marker Setup:** Once connected to the OPC UA server, the address space is explored, and for each ARObj, a corresponding object is instantiated at its respective position (this information is also available in the server’s node data). For each node, a new GameObject is instantiated. By examining the Unity Hierarchy panel in Figure 33, one can observe a series of Sphere objects with their associated CanvasOBJ, directly connected as child nodes.

The canvas was designed and saved as a prefab, allowing it to be instantiated multiple times—one for each object. This approach enables developers to easily customize the canvas by modifying the prefab, with changes propagating throughout the entire application. In this phase, the only difference from the final application on the headset is the placement of augmented markers (the spheres in this case, which are also customizable). These markers will later be positioned based on the mechanical system’s origin coordinates, obtained using Vuforia.

- **Real-Time Communication:** In this phase, each object handles user input, sends parameter changes to the server, and cyclically updates the state values displayed to the user.

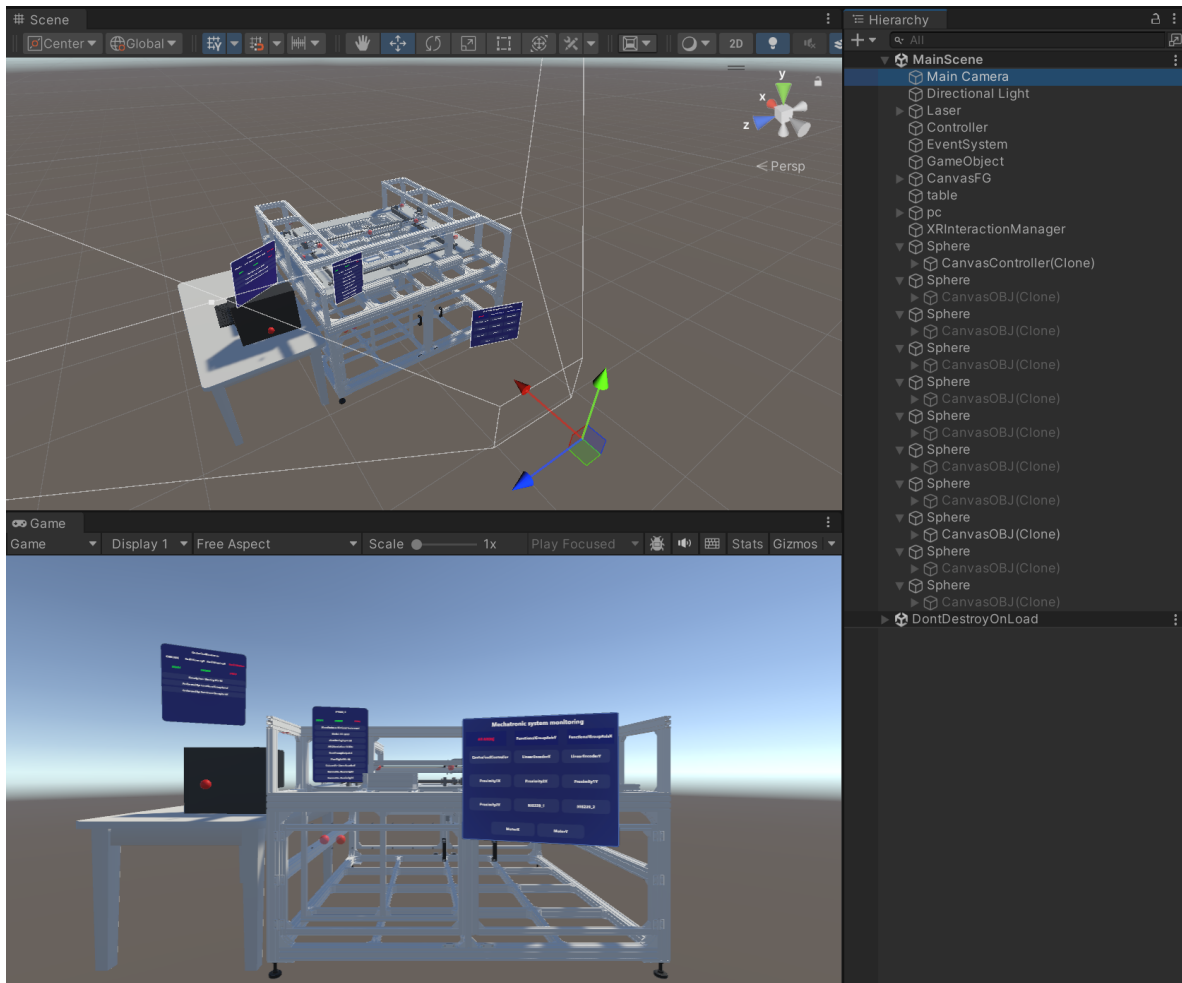


Figure 33: Unity interface showing the developed application

In summary, except for object tracking and user input handling, the entire application logic has been implemented. This simulation approach provides an excellent way to test functionalities and evaluate

application performance, as it is far simpler than debugging methods based on the logging tools available for the Magic Leap 2 headset.

Another significant advantage of this approach lies in its potential to serve as a digital twin of the system, enabling more efficient remote monitoring capabilities.

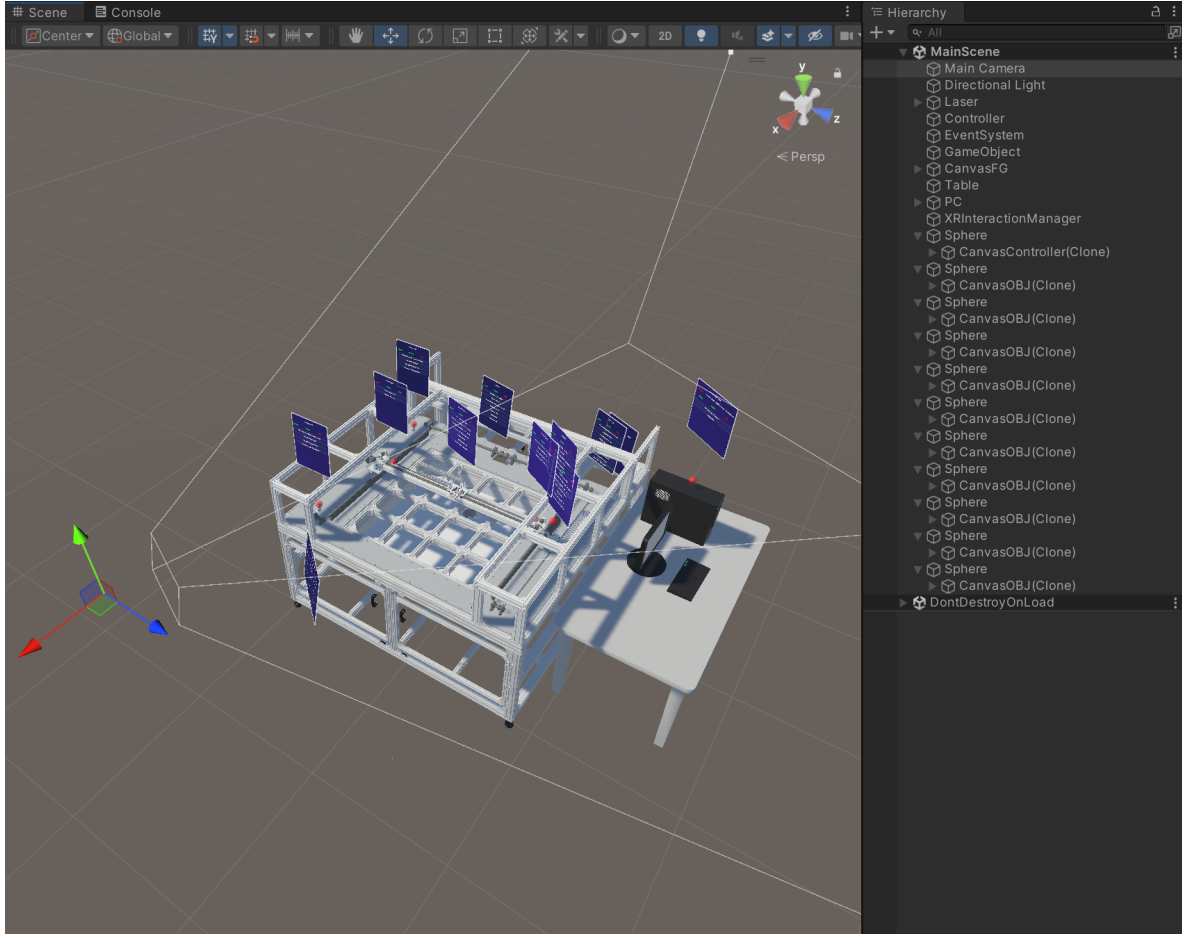


Figure 34: Scene view with all canvases enabled

In Figure 34, the 3D model of the Cartesian robot is shown along with all instantiated AR objects, including their visible canvas interfaces. The user can enable or disable these canvases by interacting with the AR marker—represented in this case by a sphere—or by using a central canvas. This central canvas follows the operator’s movements, staying out of the user’s central field of view to minimize distractions. It provides a list of all available objects and organizes them into the system’s functional blocks.

Figures 35 and 36 illustrate the aforementioned central canvas. Specifically, Figure 35 shows the Scene View, with the main camera (representing the user’s perspective) in two different configurations, and the central canvas consistently positioned within the user’s view. The canvas is oriented to ensure clear visibility and usability. Figure 36 displays the Game View corresponding to the two configurations from Figure 35. On the left, the canvas shows the list of interactable objects, while on the right, the operator has selected the functional group for the Y-axis. This view organizes the components of the

Y-axis by category, enabling the operator to access and interact with them efficiently.

Figure 35: Scene view with the main Canvas

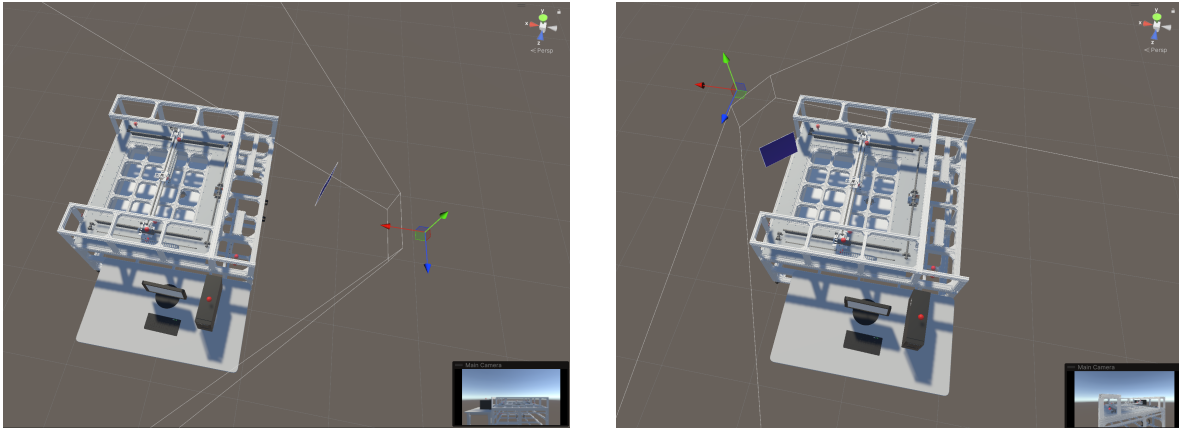
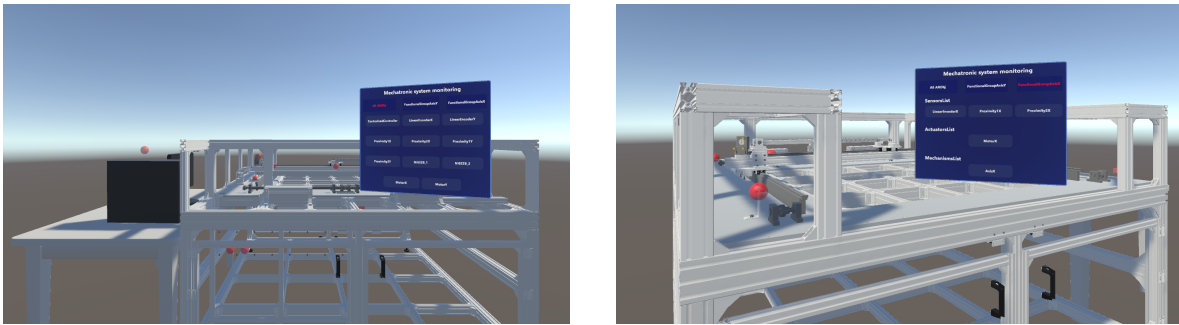


Figure 36: Game view of Figure 35 with main Canvas



The proposed information model enables not only monitoring the system components but also interacting with them and their processes. To achieve this, ARTask objects have been defined and organized within the software components of a specific ARObjct, the controller.

The application is designed to automatically detect a controller object and instantiate a dedicated user interface. This interface allows users to view the status, properties, and parameters of the system's tasks. Figure 37 provides an example of this canvas, which is accessible at the physical controller of the system, represented in this case by an industrial computer.

The image specifically shows two parameters that allow the user to issue start or stop commands for the task controlling the system's movement. Although this is a simple example, it demonstrates the potential of the adopted data structure: it enables the mapping of a wide range of variables, allowing for the complete configuration of the system's behavior. In the GENERAL tab, users can also access information about the ARObjct Controller.



Figure 37: Centralized Controller User Interface, parameters

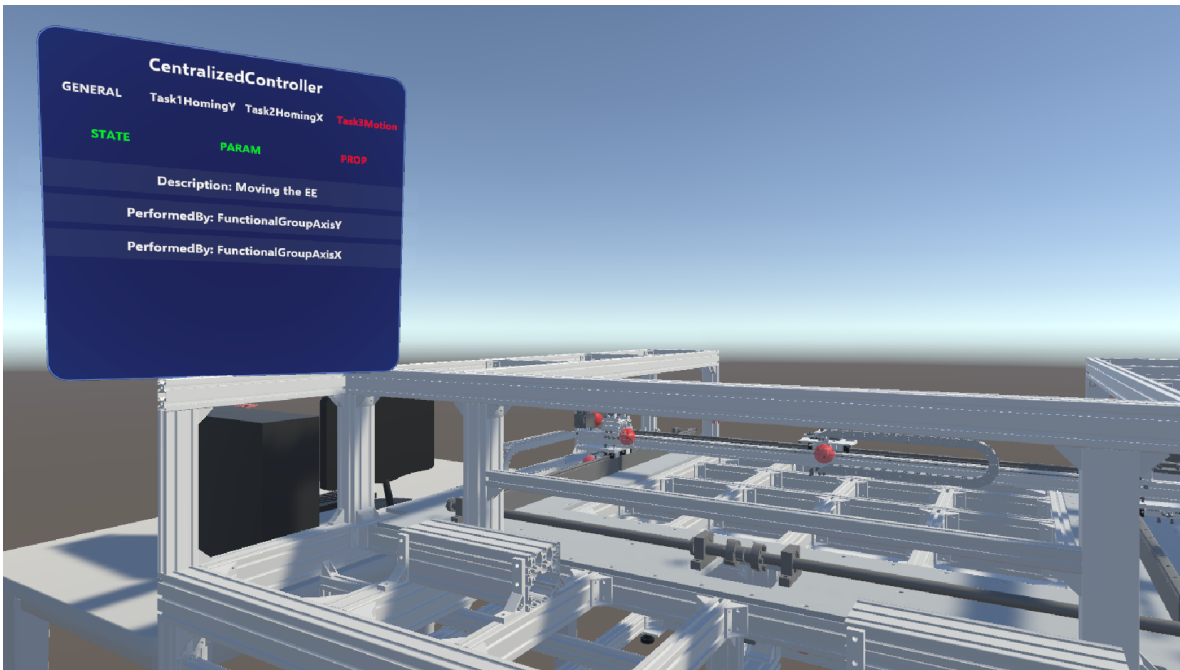


Figure 38: Centralized Controller User Interface, properties

In Figure 38, the same interface is shown from a different user perspective, demonstrating how the simulator always orients active canvases toward the observer. In this view, the operator is inspecting the properties of Task3, including both intrinsic and functional attributes. The functional attributes, in this case, define the functional blocks responsible for completing the task.

This feature of the proposed information modeling system, combined with the ability to use the main canvas to visualize components organized by functional blocks, establishes a direct link between component monitoring and process monitoring. This connection is essential for effective problem identification.

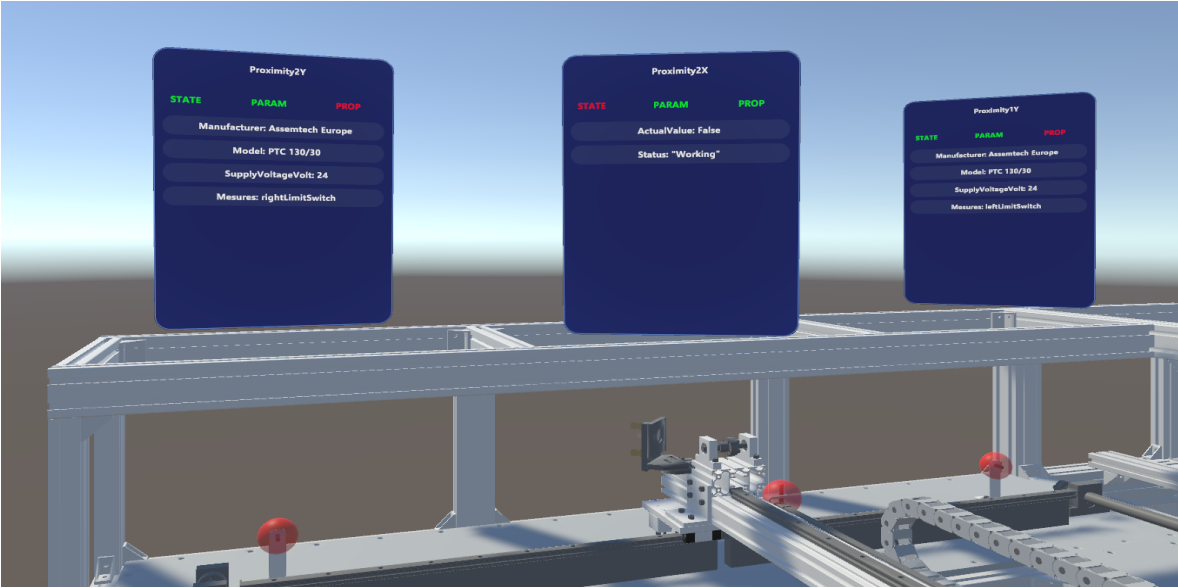


Figure 39: Details on the position of the AR markers

Figures 39, 40, and 41 provide detailed views of the interfaces for individual components, showing the represented information and the precision with which the markers are positioned to outline the object. It is worth emphasizing that, while the same level of accuracy is maintained in the AR application on the Magic Leap 2, the method used to acquire data from the headset introduces some distortion. This distortion makes it impossible to accurately display the achieved positioning.

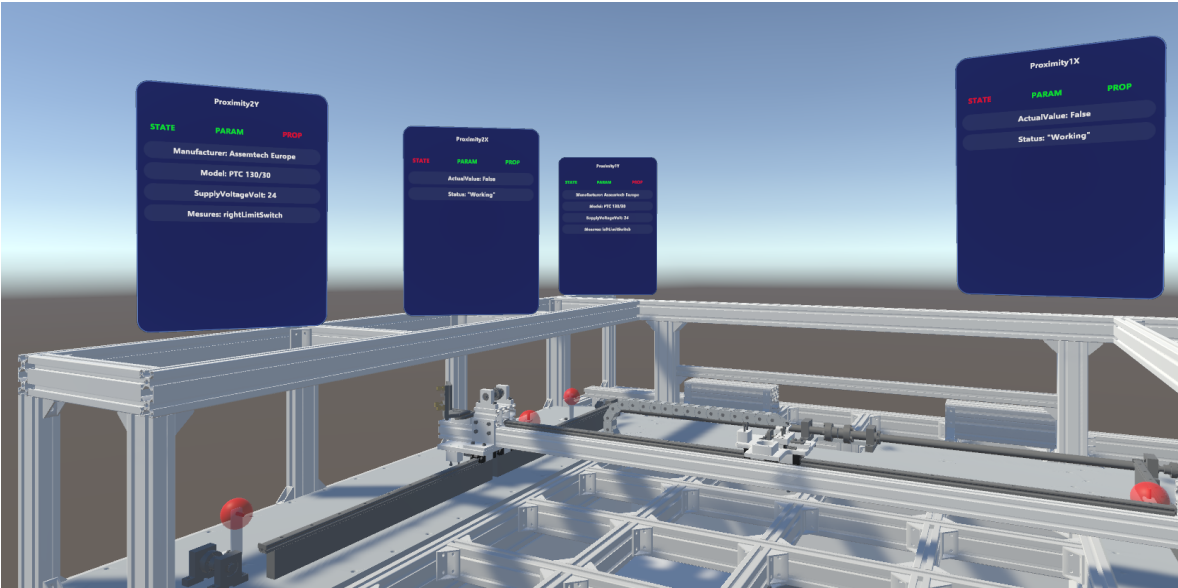


Figure 40: Details on the proximity sensors

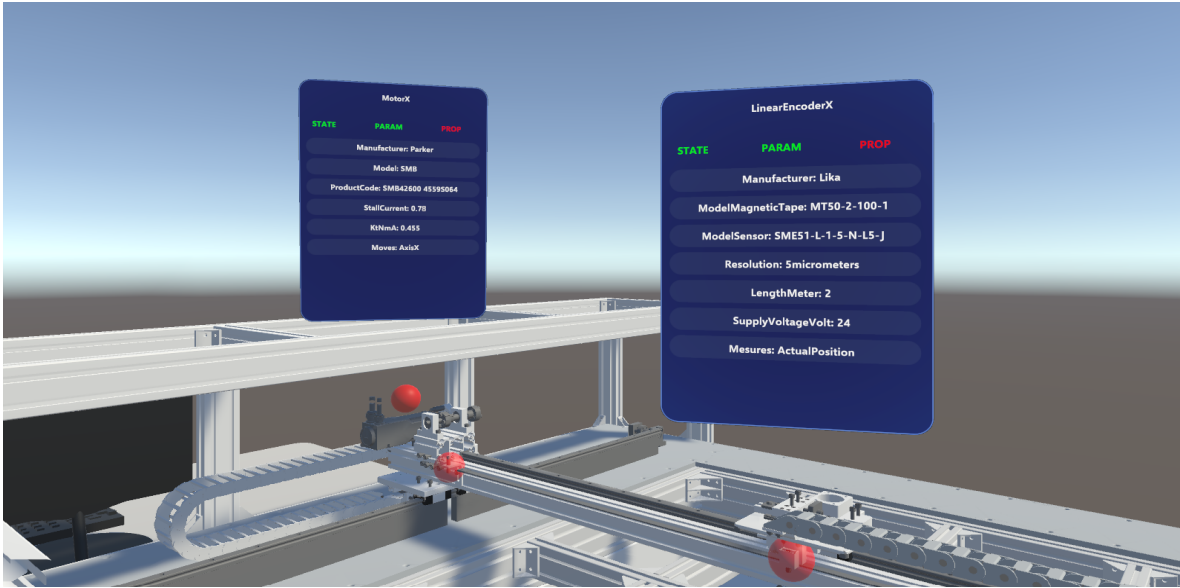


Figure 41: Linear encoder and motor details

At this point, after thoroughly describing the application as developed in the simulation environment, we can move on to presenting the results achieved using the headset.

## 5.2 Results from AR Deployment on Magic Leap 2

After describing the application development in the simulated environment, the software was extended to include system tracking using Vuforia for the AR application. The purpose of this integration was to define the system's origin coordinates. At this stage, AR markers were positioned, and the graphical interfaces were adjusted accordingly. Another significant modification involved integrating the OpenXR Action Map, ensuring that user inputs—previously simulated using a mouse and keyboard—were now connected to the headset and its controller.

The results presented here were captured using the Magic Leap Hub 3, as previously introduced. This tool allows for various functionalities, including recording the user's point of view. Unfortunately, the captured output consists of images taken by the RGB cameras on the headset, overlaid with a 2D representation of the augmented information. This capture method introduces distortions and does not reflect the full 3D visualization provided by the headset's technology for displaying augmented data.

The Magic Leap 2 leverages advanced waveguide technology to project digital content into the user's field of view. This approach allows for the seamless blending of virtual elements with the real-world environment. The waveguides guide light to the user's eyes in a precise manner, enabling a stereoscopic view of digital content. This ensures that the augmented data appears as if it is anchored in the real-world space with high accuracy and depth perception.



Figure 42: Application result on Magic Leap2: front view

While the user perceives the AR markers and interfaces in the same highly accurate spatial positions observed in the simulated environment, the recorded captures exhibit distortions due to the 2D overlay. These limitations arise from the RGB camera's inability to replicate the true depth and spatial fidelity experienced through the headset.



Figure 43: Application result on Magic Leap2: front view with controller

In Figure 42, the Cartesian robot is shown along with the user interfaces for the components and the main menu, highlighting the objects currently visible. The AR application allows the user to reposition the main menu to a convenient location by selecting it with the virtual ray that indicates the controller pointer. This feature enhances the application’s usability by enabling the user to adjust their view for comfortable operation.

In Figure 43, the controller is also displayed, along with its user interface, which provides access to the controllable tasks.

Figure 44 presents a side view, highlighting the dynamic movement of the component interfaces, which adjust to follow the operator’s position. In this view, all the canvases are enabled, showing the robot populated with all available information.



Figure 44: Application result on Magic Leap2:side view

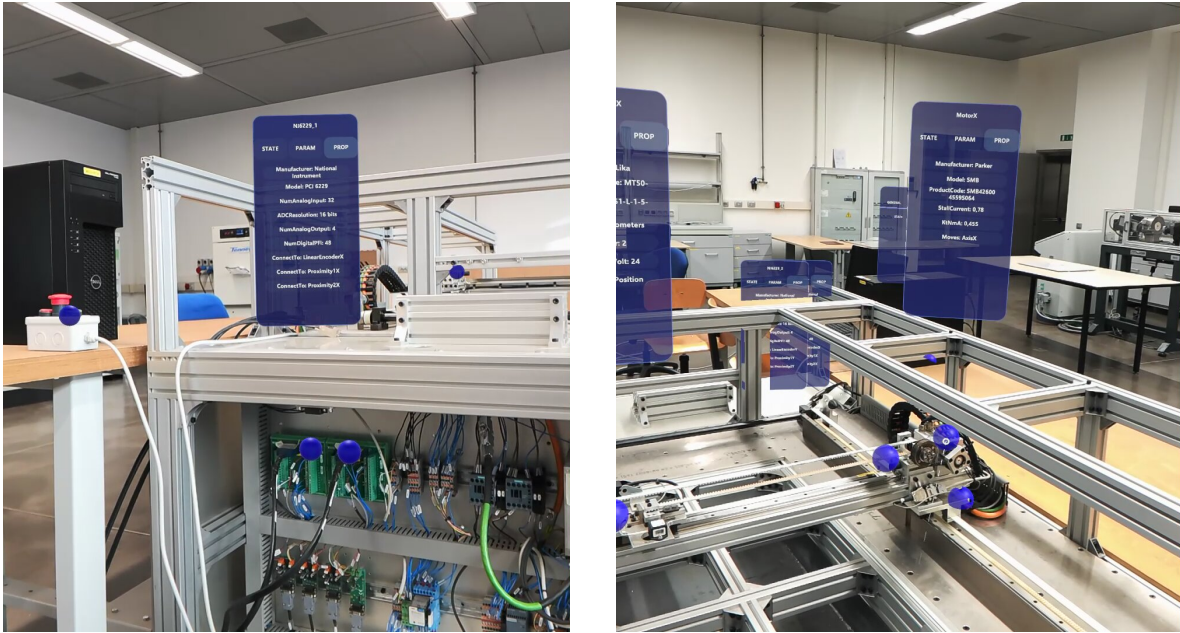
Thanks to the dynamic dimming function of Magic Leap 2, the user can see the generated interfaces with high clarity. This function works by adjusting the brightness of the virtual content based on the surrounding environment. It allows the system to automatically reduce the intensity of augmented content in well-lit environments, preventing the user from being overwhelmed by overly bright elements. As a result, the interfaces appear with optimal visibility regardless of ambient lighting conditions.

In the captures made with the Magic Leap Hub 3, it is possible to adjust the opacity of the augmented content. For this reason, the appearance of the content may vary across different captures, as the opacity is modified to account for the varying brightness of each scene. The adjustments are made to ensure the augmented information remains visible and comfortable to view in different lighting conditions.

In Figure 45, the properties of one of the two acquisition boards can be seen in detail, along with the corresponding AR marker inside the electrical panel on the wired block for signal acquisition, and those of the X-Axis motor. In the second image, the virtual beam of the pointer is also visible, used

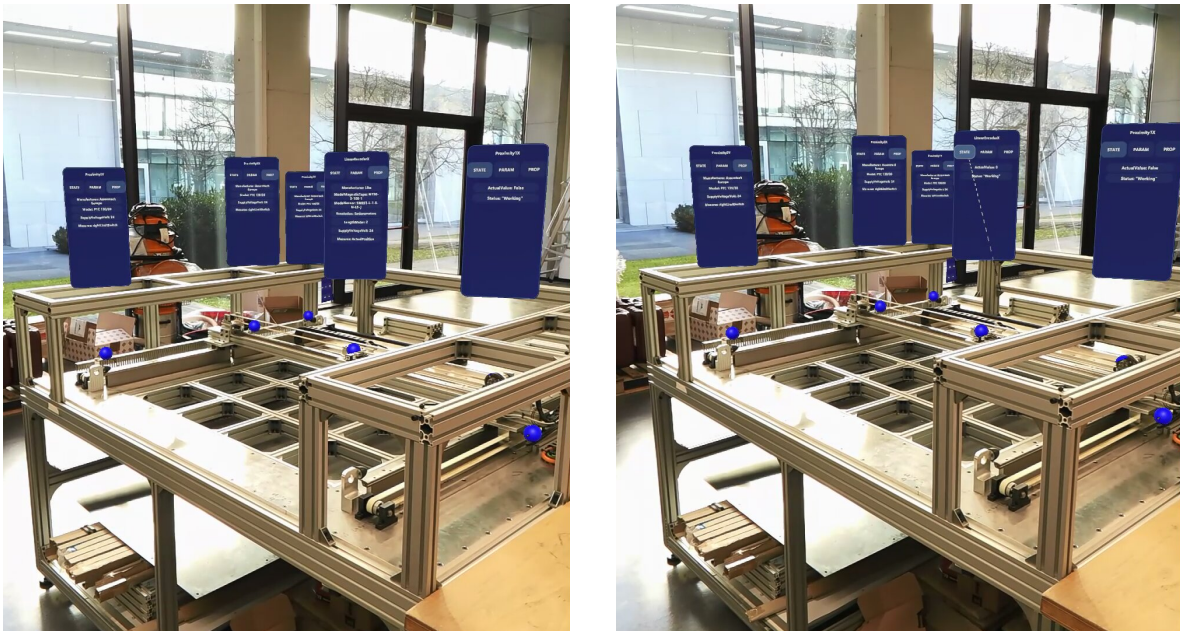
to select which information card to view.

Figure 45: NI 6229 and motor details



With yet another perspective, in Figure 46, the user is shown switching from the display of the linear encoder properties of the X-axis to the status view by selecting the corresponding button in the user interface.

Figure 46: Linear encoder State Tab selection

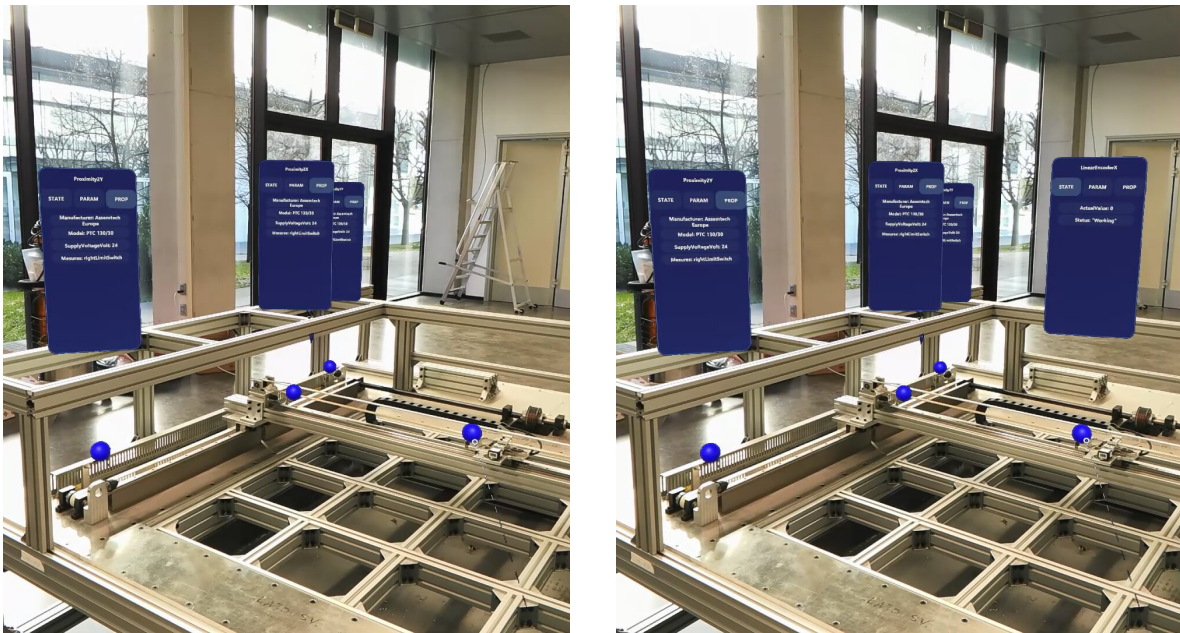


Regarding the controller, in Figure 47, the operator is shown selecting the task they wish to control and then enabling the "start" parameter, which, being boolean, is represented as a toggle button.

Figure 47: Starting Task3 using the controller UI



Figure 48: Interacting with the AR marker: Activating the X-axis linear encoder canvas with the controller



To access the menus of individual components, as mentioned earlier, users can interact with the AR marker, in this case, a sphere, as shown in Figure 48, or use the main menu, as seen in Figure 49. In Figure 48, the sequence of images shows the operator activating the display of the canvas related to the X-axis linear encoder by interacting with the marker using the controller.

In Figure 49, the main menu is visible, displaying the section for all available components, and the user enables the display of the second acquisition tab.

Figure 49: Using the main menu: Enabling the second acquisition board UI in the components section



This series of annotated images has demonstrated the functionalities of the developed application, validating the proposed framework. By following the defined guidelines—such as the procedure for designing components, the model for defining information, and the guidelines for creating the AR application—a fully operational system has been built, offering numerous advantages.

The application is system-independent, leveraging the schema used to represent the information, making it adaptable to various use cases. Additionally, due to its technical implementation, the application is device-agnostic and can function on any headset. If transitioning to a handheld device, such as a tablet, only the user inputs would need to be adjusted via the action maps discussed in the implementation chapter, defining new interaction methods with objects.

Another advantage is the high level of customization. Given the extensive research on user-related factors for designing more effective interfaces, the application allows users to modify menus simply by providing the prefab for the desired interface. While this wasn't the primary focus of the work, the proposed solution is fully compatible with integration into studies specifically targeting this area.

The next and final chapter will present the conclusions on the proposed framework, highlighting its strengths and potential future developments.

## 6 Conclusions

The thesis work presented in this document aims to develop a framework, understood as a set of conceptual methods, to facilitate the creation of real-time human-machine interaction systems leveraging augmented reality (AR). Specifically, interaction refers to the user's ability to utilize augmented reality to access information about system components and processes, enabling the configuration and control of desired functionalities.

The literature review highlights that this technology is an effective tool in supporting industrial development under the Industry 5.0 paradigm, promoting a human-centric approach with a focus on sustainability. However, the lack of studies proposing a structured method for developing such systems hinders their widespread adoption and the full exploitation of their capabilities. Based on these premises, the proposed framework addresses the primary challenges associated with integrating AR into industrial contexts. These challenges include integrating AR with existing systems, representing the large volumes of heterogeneous data characterizing a mechatronic system, and developing the application itself.

The proposed framework outlines a procedure to support system design, providing clear steps and the necessary information for each phase, ultimately leading to a comprehensive system definition. This procedure allows for the identification of requirements and specifications for the AR application, enabling the determination of the optimal hardware and software for the component. It also facilitates the representation of the information related to the mechatronic system with which the application will interact, offering a well-defined structure based on an object-oriented representation of the system.

The model relies on the concept of functional groups, enabling not only the representation of system components but also the relationships between them, thereby enhancing the data's value for the operator. Recalling that the data required by the procedure includes information on the system and its 3D model, the procedure then defines the intermediate element - the server - that acts as a bridge between the mechatronic system and the AR application, thus overcoming the challenges of integration.

Finally, the framework proposes a structure for application development, leveraging the proposed data model, allowing for a single application compatible with all systems using the same data representation model.

The framework was validated using a Cartesian robot at the Mechatronic and Mechanical Dynamics Lab of the University and a Magic Leap 2 headset. All steps outlined in the conceptual chapter were applied to this case study. The results showcase an application that visualizes all system components modeled as proposed, with real-time updates on their properties and information. The operator can interact with system tasks, initiating specific operations.

This case study highlights the potential of the proposed framework. Future developments could apply the framework to more complex scenarios, demonstrating its flexibility and modularity. Another avenue for exploration could be integrating this study with user-related research, testing the application, and enhancing its interface to better suit user needs.

Another potential development involves the concept of digital twins. The application can be simu-

lated on the system's 3D model, enabling remote monitoring through the representation of the system's state. Adding the capability to animate sub-parts of the 3D model in line with existing data could lead to a comprehensive digital twin, integrating both kinematics and dynamics of the system.

In conclusion, this work highlights several innovative aspects: firstly, the definition of a well-structured procedure, a data representation model to support the augmented context, a well defined workflow for developing the AR application, and a detailed vision of how all the conceptual proposals can be effectively applied.

## References

- [1] Chiara Belsito, Chiara Franciosi, and Valentina Di Pasquale. The impact of augmented reality devices on operator performance in manufacturing contexts. *Management and Production Engineering Review*, 15:12–34, 3 2024.
- [2] Muhammad Hamza Zafar, Even Falkenberg Langås, and Filippo Sanfilippo. Exploring the synergies between collaborative robotics, digital twins, augmentation, and industry 5.0 for smart manufacturing: A state-of-the-art review. *Robotics and Computer-Integrated Manufacturing*, 89:102769, 10 2024.
- [3] Wei Fang, Lixi Chen, Tienong Zhang, Chengjun Chen, Zhan Teng, and Lihui Wang. Head-mounted display augmented reality in manufacturing: A systematic review, 10 2023.
- [4] Carlos E. Mendoza-Ramírez, Juan C. Tudon-Martinez, Luis C. Félix-Herrán, Jorge de J. Lozoya-Santos, and Adriana Vargas-Martínez. Augmented reality: Survey, 9 2023.
- [5] William Z. Bernstein, Andrew Bowman, Ryan Durscher, Andrew Gillman, and Sean Donegan. Towards data and model interoperability for industrial extended reality in manufacturing. *Journal of Computing and Information Science in Engineering*, 23, 12 2023.
- [6] A.Y.C. Nee, S.K. Ong, G. Chryssolouris, and D. Mourtzis. Augmented reality applications in design and manufacturing. *CIRP Annals*, 61(2):657–679, 2012.
- [7] Paula Fraga-Lamas, Tiago M. Fernández-Caramés, Óscar Blanco-Novoa, and Miguel A. Vilar-Montesinos. A review on industrial augmented reality systems for the industry 4.0 shipyard. *IEEE Access*, 6:13358–13375, 2018.
- [8] Riccardo Palmarini, John Ahmet Erkoyuncu, Rajkumar Roy, and Hosein Torabmostaedi. A systematic review of augmented reality applications in maintenance. *Robotics and Computer-Integrated Manufacturing*, 49:215–228, 2018.
- [9] Eleonora Bottani and Giuseppe Vignali. Augmented reality technology in the manufacturing industry: A review of the last decade. *IIE Transactions*, 51:284–310, 03 2019.
- [10] Johannes Egger and Tariq Masood. Augmented reality in support of intelligent manufacturing – a systematic literature review. *Computers and Industrial Engineering*, 140:106195, 2020.
- [11] Dawi Karomati Baroroh, Chih-Hsing Chu, and Lihui Wang. Systematic literature review on augmented reality in smart manufacturing: Collaboration between human and computational intelligence. *Journal of Manufacturing Systems*, 61:696–711, 2021.
- [12] Luís Fernando de Souza Cardoso, Flávia Cristina Martins Queiroz Mariano, and Ezequiel Roberto Zorzal. A survey of industrial augmented reality. *Computers and Industrial Engineering*, 139:106159, 2020.
- [13] Gheorghe Daniel Voinea, Florin Gîrbacia, Mihai Duguleană, Răzvan Gabriel Boboc, and Carmen Gheorghe. Mapping the emergent trends in industrial augmented reality, 4 2023.
- [14] M. Eswaran, A.K. Gulivindala, A.K. Inkulu, and M.V.A. Raju Bahubalendruni. Augmented reality-based guidance in product assembly and maintenance/repair perspective: A state of the art review on challenges and opportunities. *Expert Systems with Applications*, 213, 2023.

- [15] Ahmed A.A.E.M.A. Khafaga, Lorena Caires Moreira, and Ben Horan. Towards industry 5.0: Augmented reality assistance systems for people-centred digitalisation and smart manufacturing. In *ICAC 2023 - 28th International Conference on Automation and Computing*. Institute of Electrical and Electronics Engineers Inc., 2023.
- [16] Ana Malta, Torres Farinha, and Mateus Mendes. Augmented reality in maintenance—history and perspectives. *Journal of Imaging*, 9, 7 2023.
- [17] Yue Yin, Pai Zheng, Chengxi Li, and Lihui Wang. A state-of-the-art survey on augmented reality-assisted digital twin for futuristic human-centric industry transformation, 6 2023.
- [18] Akekathed Sanglub, Prachyanun Nilsook, and Panita Wannapiroon. Imagineering on augmented reality and digital twin for digital competence. *International Journal of Information and Education Technology*, 9:213–217, 01 2019.
- [19] Chan Qiu, Shien Zhou, Zhenyu Liu, Qi Gao, and Jianrong Tan. Digital assembly technology based on augmented reality and digital twins: a review. *Virtual Reality and Intelligent Hardware*, 1(6):597–610, 2019. VR/AR in industry.
- [20] Corentin Coupry, Sylvain Noblecourt, Paul Richard, David Baudry, and David Bigaud. Bim-based digital twin and xr devices to improve maintenance procedures in smart buildings: A literature review. *Applied Sciences*, 11(15), 2021.
- [21] Nina Tvenge, Olga Ogorodnyk, Niels Peter Østbø, and Kristian Martinsen. Added value of a virtual approach to simulation-based learning in a manufacturing learning factory. *Procedia CIRP*, 88:36–41, 2020. 13th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 17-19 July 2019, Gulf of Naples, Italy.
- [22] Samad ME Sepasgozar, Mohsen Ghobadi, Sara Shirowzhan, David J Edwards, and Elham Delzendeh. Metrics development and modelling the mixed reality and digital twin adoption in the context of industry 4.0. *Engineering, Construction and Architectural Management*, 28(5):1355–1376, 2021.
- [23] YK Liu, SK Ong, and AYC Nee. State-of-the-art survey on digital twin implementations. *Advances in Manufacturing*, 10(1):1–23, 2022.
- [24] Andreas Künz, Sabrina Rosmann, Enrica Loria, and Johanna Pirker. The potential of augmented reality for digital twins: A literature review. In *2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 389–398, 2022.
- [25] Joel Murithi Runji, Yun Ju Lee, and Chih Hsing Chu. Systematic literature review on augmented reality-based maintenance applications in manufacturing centered on operator needs, 3 2023.
- [26] Phuong Thao Ho, José Antonio Albajez, Jorge Santolaria, and José A Yagüe-Fabra. Study of augmented reality based manufacturing for further integration of quality control 4.0: A systematic literature review. *Applied Sciences*, 12(4):1961, 2022.
- [27] Nuno Cid Martins, Bernardo Marques, João Alves, Tiago Araújo, Paulo Dias, and Beatriz Sousa Santos. Augmented reality situated visualization in decision-making. *Multimedia Tools and Applications*, 81:14749–14772, 5 2022.

- [28] Gabriel de Moura Costa, Marcelo Roberto Petry, and António Paulo Moreira. Augmented reality for human–robot collaboration and cooperation in industrial applications: A systematic literature review. *Sensors*, 22, 4 2022.
- [29] Tariq Masood and Johannes Egger. Augmented reality in support of industry 4.0—implementation challenges and success factors. *Robotics and Computer-Integrated Manufacturing*, 58:181–195, 2019.
- [30] Sebastian Blankemeyer, David Wendorff, and Annika Raatz. A hand-interaction model for augmented reality enhanced human-robot collaboration. *CIRP Annals*, 1 2024.
- [31] Sri Sudha Vijay Keshav Kolla and Peter Plapper. Interaction modalities for augmented reality applications in manufacturing. In *Advances in Transdisciplinary Engineering*, volume 35, pages 379–390. IOS Press BV, 7 2023.
- [32] Ana C. Pereira, Anabela C. Alves, and Pedro Arezes. Augmented reality in a lean workplace at smart factories: A case study. *Applied Sciences (Switzerland)*, 13, 8 2023.
- [33] Alexandre Carrana, Nuno Sousa, Jose Rocha, Eduardo Santos, Luis Evangelista, Antonio Ferreira, Telmo Adao, Emanuel Sousa, and Iara Margolis. Augmented reality towards industry 5.0: Improving voice and tap interaction based on user experience feedback. In *Lecture Notes in Electrical Engineering*, volume 1076 LNEE, pages 160–171. Springer Science and Business Media Deutschland GmbH, 2024.
- [34] Iara Margolis, Alexandre Carrana, Nuno Sousa, Jose Rocha, Eduardo Santos, Luis Evangelista, Antonio Ferreira, Telmo Adao, and Emanuel Sousa. Ux and industry 5.0: A study in repairing equipment using augmented reality. In *Lecture Notes in Electrical Engineering*, volume 1076 LNEE, pages 146–159. Springer Science and Business Media Deutschland GmbH, 2024.
- [35] Michele Gattullo, Giulia Wally Scurati, Michele Fiorentino, Antonio Emmanuele Uva, Francesco Ferrise, and Monica Bordegoni. Towards augmented reality manuals for industry 4.0: A methodology. *Robotics and Computer-Integrated Manufacturing*, 56:276–286, 4 2019.
- [36] Liru Chen, Hantao Zhao, Chenhui Shi, Youbo Wu, Xuwen Yu, Wenzhe Ren, Ziyi Zhang, and Xiaomeng Shi. Enhancing multi-modal perception and interaction: An augmented reality visualization system for complex decision making. *Systems*, 12, 1 2024.
- [37] Jinfang Li, Yongxi Liu, Libao Xiao, Jiahong Cai, Hanwu He, and Xiaowei Zhang. Augmented reality creation platform based on graphical programming. *IEEE Access*, 12:14455–14465, 2024.
- [38] Santina Fortuna, Loris Barbieri, Emanuele Marino, and Fabio Bruno. A comparative study of augmented reality rendering techniques for industrial assembly inspection. *Computers in Industry*, 155:104057, 2 2024.
- [39] Emanuele Marino, Loris Barbieri, Fabio Bruno, and Maurizio Muzzupappa. Context-aware recoloring approach for ar-assisted maintenance applications. In *Lecture Notes in Mechanical Engineering*, pages 471–478. Springer Science and Business Media Deutschland GmbH, 2024.
- [40] Rafael Maio, Tiago Araujo, Bernardo Marques, Andre Santos, Pedro Ramalho, Duarte Almeida, Paulo Dias, and Beatriz Sousa Santos. Pervasive augmented reality to support real-time data monitoring in industrial scenarios: Shop floor visualization evaluation and user study. *Computers-Graphics*, 118:11–22, 2 2024.

- [41] Rainer Eber, Dennis Kollmann, Doris Aschenbrenner, Maximilian Hentsch, Steffen Schwarzer, and Nicole Stricker. Iiot visualization applications based on augmented reality - practical approach for easy implementation. In *Procedia CIRP*, volume 120, pages 964–967. Elsevier B.V., 2023.
- [42] Jay P. Patel, Sanket R. Iyer, and Sarosh K. Dastoor. Predictive maintenance of machine health by s-node using ar. In *Lecture Notes in Networks and Systems*, volume 833, pages 311–322. Springer Science and Business Media Deutschland GmbH, 2024.
- [43] Andreas Jakl, Lucas Schöffner, Matthias Husinsky, and Markus Wagner. Augmented reality for industry 4.0: Architecture and user experience. 11 2018.
- [44] T. Gramberg, K. Kruger, and J. Niemann. *Augmented Reality for Operators in Smart Manufacturing Environments: A Case Study Implementation*, volume Part F1162, pages 401–413. Springer Nature, 2023.
- [45] Filip Žemla, Ján Cigánek, Danica Rosinová, Erik Kučera, and Oto Haffner. Smart platform for monitoring and control of discrete event system in industry 4.0 concept. *Applied Sciences (Switzerland)*, 13, 10 2023.
- [46] Dony Novalindry, Rahmat Febri Yoga Saputra, Novi Febrianti, Doni Tri Putra Yanto, Fadhillah Majid Saragih, and Wan Mohd Yusof Rahiman. Development of a digital twin prototype for industrial manufacturing monitoring system using iot and augmented reality. *International journal of online and biomedical engineering*, 20:4–23, 2024.
- [47] Thomas Schmitt, Philip Viklund, Martina Sjölander, Lars Hanson, Kaveh Amouzgar, and Matías Urenda Moris. Augmented reality for machine monitoring in industrial manufacturing: framework and application development. *Procedia CIRP*, 120:1327–1332, 1 2023.
- [48] Erich Stark, Erik Kučera, Oto Haffner, Peter Drahoš, and Roman Leskovský. Using augmented reality and internet of things for control and monitoring of mechatronic devices. *Electronics*, 9(8), 2020.
- [49] San Giliyana, Joakim Karlsson, Marcus Bengtsson, Antti Salonen, Vincent Adoue, and Mikael Hedelind. A testbed for smart maintenance technologies. In *Lecture Notes in Mechanical Engineering*, pages 437–450. Springer Science and Business Media Deutschland GmbH, 2024.
- [50] Changchun Liu, Haihua Zhu, Dunbing Tang, Qingwei Nie, Tong Zhou, Liping Wang, and Yejia Song. Probing an intelligent predictive maintenance approach with deep learning and augmented reality for machine tools in iot-enabled manufacturing. *Robotics and Computer-Integrated Manufacturing*, 77, 10 2022.
- [51] Vachara Peansupap, Aye Chann Myint, and Kriengsak Panuwatwanich. Application of augmented reality for checking scaffolding installation in maintenance process. In *Lecture Notes in Civil Engineering*, volume 369, pages 743–752. Springer Science and Business Media Deutschland GmbH, 2024.
- [52] János Simon, László Gogolák, József Sárosi, and Igor Fürstner. Augmented reality based distant maintenance approach. *Actuators*, 12, 7 2023.

- [53] Doris Aschenbrenner, Marc Erich Latoschik, and Klaus Schilling. Industrial maintenance with augmented reality: two case studies. In *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology*, pages 341–342, 2016.
- [54] James Frandsen, Joe Tenny, Walter Frandsen, and Yuri Hovanski. An augmented reality maintenance assistant with real-time quality inspection on handheld mobile devices. *International Journal of Advanced Manufacturing Technology*, 125:4253–4270, 4 2023.
- [55] Lorenzo Valentini, Fabio Grandi, Margherita Peruzzini, and Marcello Pellicciari. Ux-driven methodology to design usable augmented reality applications for maintenance. In *Advances in Transdisciplinary Engineering*, volume 41, pages 42–51. IOS Press BV, 11 2023.
- [56] Ana Cachada, David Costa, Hasmik Badikyan, José Barbosa, Paulo Leitão, Osmano Morais, Carlos Teixeira, João Azevedo, Pedro Miguel Moreira, and Luís Romero. Using ar interfaces to support industrial maintenance procedures. In *IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society*, volume 1, pages 3795–3800, 2019.
- [57] Susanna Aromaa, Antti Väättänen, Mika Hakkarainen, and Eija Kaasinen. User experience and user acceptance of an augmented reality based knowledge-sharing solution in industrial maintenance work. In Tareq Ahram and Christianne Falcão, editors, *Advances in Usability and User Experience*, pages 145–156, Cham, 2018. Springer International Publishing.
- [58] John Angelopoulos and Dimitris Mourtzis. An intelligent product service system for adaptive maintenance of engineered-to-order manufacturing equipment assisted by augmented reality. *Applied Sciences (Switzerland)*, 12, 6 2022.
- [59] N. Tobiskova, M. Hattinger, and E. Sanderson Gull. Evaluating an augmented reality prototype for enhanced user guidance in an industrial production context. In *Advances in Transdisciplinary Engineering*, volume 52, pages 419–430, 2024.
- [60] Enricoandrea Laviola, Sara Romano, Michele Gattullo, and Antonio Emmanuele Uva. Virtual mirror for maintenance: An augmented reality support tool. In *Lecture Notes in Mechanical Engineering*, pages 512–520. Springer Science and Business Media Deutschland GmbH, 2024.
- [61] Zhengjie Xue, Jun Yang, Ruchen Chen, Qiang He, Qixiu Li, and Xuesong Mei. Ar-assisted guidance for assembly and maintenance of avionics equipment. *Applied Sciences (Switzerland)*, 14, 2 2024.
- [62] Justyna Trojanowska, Jakub Kašćak, Jozef Husár, and Lucia Knapčíková. Possibilities of increasing production efficiency by implementing elements of augmented reality. *Bulletin of the Polish Academy of Sciences: Technical Sciences*, 70, 2022.
- [63] Riccardo Palmarini, Inigo Fernandez Del Amo, Dedy Ariansyah, Samir Khan, John Ahmet Erkoyuncu, and Rajkumar Roy. Fast augmented reality authoring: Fast creation of ar step-by-step procedures for maintenance operations. *IEEE Access*, 11:8407–8421, 2023.
- [64] Grégoire Mompeu, Florence Danglade, Frédéric Mérienne, and Christophe Guillet. Methodology for augmented reality-based adaptive assistance in industry. *Computers in Industry*, 154, 1 2024.
- [65] Philipp Klimant and Christian Kollatsch. Concepts for creating augmented reality based technical documentations for the maintenance of machine tools. *International Journal on Interactive Design and Manufacturing*, 16:765–773, 6 2022.

- [66] Christian Kollatsch and Philipp Klimant. Efficient integration process of production data into augmented reality based maintenance of machine tools. *Production Engineering*, 15(3-4):311 – 319, 2021. Cited by: 11; All Open Access, Hybrid Gold Open Access.
- [67] John Ahmet Erkoyuncu, Iñigo Fernández del Amo, Michela Dalle Mura, Rajkumar Roy, and Gino Dini. Improving efficiency of industrial maintenance with context aware adaptive authoring in augmented reality. *CIRP Annals*, 66(1):465–468, 2017.
- [68] Arne Seeliger, Long Cheng, and Torbjørn Netland. Augmented reality for industrial quality inspection: An experiment assessing task performance and human factors. *Computers in Industry*, 151, 10 2023.
- [69] Adam Nowak, Yuchong Zhang, Andrzej Romanowski, and Morten Fjeld. Augmented reality with industrial process tomography: To support complex data analysis in 3d space. In *Adjunct Proceedings of the 2021 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2021 ACM International Symposium on Wearable Computers*, UbiComp/ISWC '21 Adjunct, page 56–58, New York, NY, USA, 2021. Association for Computing Machinery.
- [70] Loris Barbieri, Emanuele Marino, Fabio Bruno, Maurizio Muzzupappa, and Biagio Colacino. User-centered design of an augmented reality inspection tool for industry 4.0 operators. *International Journal on Interactive Design and Manufacturing*, 2024.
- [71] Yanzhang Tong, Qiyuan Zhang, Francisco Munguia Galeano, and Ze Ji. Ar and hrc integration for enhanced pragmatic quality.
- [72] Sotiris Makris, Panagiotis Karagiannis, Spyridon Koukas, and Aleksandros Stereos Matthaiakis. Augmented reality system for operator support in human–robot collaborative assembly. *CIRP Annals - Manufacturing Technology*, 65:61–64, 2016.
- [73] George Michalos, Panagiotis Karagiannis, Sotiris Makris, Önder Tokçalar, and George Chrysolouris. Augmented reality (ar) applications for supporting human-robot interactive cooperation. In *Procedia CIRP*, volume 41, pages 370–375. Elsevier B.V., 2016.
- [74] Bernard Schmidt, Adrian Sánchez De Ocãna Torroba, Göran Grahn, Ingemar Karlsson, and Amos H.C. Ng. Augmented reality approach for a user interface in a robotic production system. In *Advances in Transdisciplinary Engineering*, volume 21, pages 240–251. IOS Press BV, 4 2022.
- [75] Sonia Mary Chacko and Vikram Kapila. Augmented reality as a medium for human-robot collaborative tasks. In *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pages 1–8. IEEE, 2019.
- [76] Kishan Chandan, Vidisha Kudalkar, Xiang Li, and Shiqi Zhang. Negotiation-based human-robot collaboration via augmented reality, 2020.
- [77] Xi Vincent Wang and Lihui Wang. *Augmented Reality Enabled Human–Robot Collaboration*, pages 395–411. Springer International Publishing, Cham, 2021.
- [78] Julia Hertel, Solmaz Goodarzi, and Frank Steinicke. Adaptive immersion: Mixed reality map visualization with gradual transition. 2024. Cited by: 0.

- [79] Jonathan M. Bird, Philip A. Smart, David J. Harris, Laura A. Phillips, Gabriella Giannachi, and Samuel J. Vine. A magic leap in tourism: Intended and realized experience of head-mounted augmented reality in a museum context. *Journal of Travel Research*, 62(7):1427 – 1447, 2023. Cited by: 8; All Open Access, Green Open Access, Hybrid Gold Open Access.
- [80] Isabelle Verhulst, Andy Woods, Laryssa Whittaker, James Bennett, and Polly Dalton. Do vr and ar versions of an immersive cultural experience engender different user experiences? *Computers in Human Behavior*, 125, 2021. Cited by: 63; All Open Access, Hybrid Gold Open Access.
- [81] Giulia Zari, Sara Condino, Fabrizio Cutolo, and Vincenzo Ferrari. Magic leap 1 versus microsoft hololens 2 for the visualization of 3d content obtained from radiological images. *Sensors*, 23(6), 2023. Cited by: 12; All Open Access, Gold Open Access, Green Open Access.
- [82] Aviv Elor, Steve Whittaker, Sri Kurniawan, and Sam Michael. Biolumin: An immersive mixed reality experience for interactive microscopic visualization and biomedical research annotation. *ACM Transactions on Computing for Healthcare*, 3(4), 2022. Cited by: 2; All Open Access, Bronze Open Access.
- [83] Jose Valerio, Maria P. Fernandez Gomez, Arturo Ayala Arcipreste, Noe Santiago Rea, Penelope Mantilla, Immanuel O. Olarinde, and Andres M. Alvarez-Pinzon. Exploring the potential use of virtual reality with a supraorbital keyhole craniotomy for anterior skull base meningiomas: Two case reports. *Journal of Personalized Medicine*, 14(11), 2024. Cited by: 0; All Open Access, Gold Open Access.
- [84] Henrik Frisk, Gustav Burström, Juliane Weinzierl, Linda Westernhagen, Florentin Tranchant, Erik Edström, and Adrian Elmi-Terander. Pedicle screw placement using an augmented reality head-mounted display in a porcine model. *Journal of Visualized Experiments*, 2024(207), 2024. Cited by: 0.
- [85] Nawal M. Alharbi and Reham B Osman. Augmented reality assisted intraoral scanning of mandibular arch: A proof-of-concept pilot clinical study. *Journal of Dentistry*, 150, 2024. Cited by: 0.
- [86] Caio A. Neves, Yona Vaisbuch, Christoph Leuze, Jennifer A. McNab, Bruce Daniel, Nikolas H. Blevins, and Peter H. Hwang. Application of holographic augmented reality for external approaches to the frontal sinus. *International Forum of Allergy and Rhinology*, 10(7):920 – 925, 2020. Cited by: 27.
- [87] Jara S. van Bergem, Pieter F. van Doorn, Eva M. Hoogendoorn, Daphne J. Geerse, and Melvyn Roerdink. Gait and balance assessments with augmented reality glasses in people with parkinson’s disease: Concurrent validity and test–retest reliability. *Sensors*, 24(17), 2024. Cited by: 1; All Open Access, Gold Open Access.
- [88] Lotte E. S. Hardeman, Daphne J. Geerse, Eva M. Hoogendoorn, Jorik Nonnekes, and Melvyn Roerdink. Remotely prescribed, monitored, and tailored home-based gait-and-balance exergaming using augmented reality glasses: a clinical feasibility study in people with parkinson’s disease. *Frontiers in Neurology*, 15, 2024. Cited by: 0; All Open Access, Gold Open Access.
- [89] Eva M. Hoogendoorn, Daphne J. Geerse, Annejet T. van Dam, John F. Stins, and Melvyn Roerdink. Gait-modifying effects of augmented-reality cueing in people with parkinson’s disease. *Frontiers in Neurology*, 15, 2024. Cited by: 3; All Open Access, Gold Open Access.

- [90] Christian Gießer, Johannes Schmitt, Emma Lowenstein, Christian Weber, Veit Braun, and Rainer Bruck. Skillslab+ - a new way to teach practical medical skills in an augmented reality application with haptic feedback. *IEEE Transactions on Learning Technologies*, 17:2034 – 2047, 2024. Cited by: 0.
- [91] Arturo Silvero Isidre, Hendrik Friederichs, Michael Müther, Marco Gallus, Walter Stummer, and Markus Holling. Mixed reality as a teaching tool for medical students in neurosurgery. *Medicina (Lithuania)*, 59(10), 2023. Cited by: 3; All Open Access, Gold Open Access, Green Open Access.
- [92] Ramiro Serrano-Vergel, Pedro Morillo, Sergio Casas-Yrurzum, and Carolina Cruz-Neira. Exploring the suitability of using virtual reality and augmented reality for anatomy training. *IEEE Transactions on Human-Machine Systems*, 53(2):378 – 389, 2023. Cited by: 6.
- [93] Olivia Hess, Jimmy Qian, Janine Bruce, Ellen Wang, Samuel Rodriguez, Nick Haber, and Thomas J. Caruso. Communication skills training using remote augmented reality medical simulation: a feasibility and acceptability qualitative study. *Medical Science Educator*, 32(5):1005 – 1014, 2022. Cited by: 11; All Open Access, Bronze Open Access, Green Open Access.
- [94] Bioingeniería Y. Práctica Clínica, Adrian Nacarino, Anderson La-Rosa, Bryan Sanchez, Jose Cornejo, Mariela Vargas, Jorge Cornejo, and Ricardo Palomares. Mixed reality in surgical mechatronics: Innovations and challenges in bioengineering education and clinical practice; [realidad mixta en mecatrónica quirúrgica: Innovaciones y retos en la educación en bioingeniería y práctica clínica]. *Revista de la Facultad de Medicina Humana*, 24(4):215 – 223, 2024. Cited by: 0.
- [95] Axel Busboom. Automated generation of opc ua information models — a review and outlook. *Journal of Industrial Information Integration*, 39:100602, 2024.
- [96] Florian Pauker, Thomas Frühwirth, Burkhard Kittl, and Wolfgang Kastner. A systematic approach to opc ua information model design. *Procedia CIRP*, 57:321–326, 2016. Factories of the Future in the digital environment - Proceedings of the 49th CIRP Conference on Manufacturing Systems.
- [97] Sebastian Friedl, Christian von Arnim, Armin Lechler, and Alexander Verl. Generation of opc ua companion specification with eclipse modeling framework. In *2020 16th IEEE International Conference on Factory Communication Systems (WFCS)*, pages 1–7. IEEE, 2020.
- [98] Thomas Goldschmidt and Wolfgang Mahnke. Evaluating domain-specific languages for the development of opc ua based applications. *IFAC Proceedings Volumes*, 45(2):860–865, 2012.
- [99] Rongkai Wang, Shibo He, Yang Liu, Chengwei Zhou, Zhiguo Shi, and Jiming Chen. Leveraging relational database for fast construction of opc ua information model. In *2021 International Conference on Intelligent Technology and Embedded Systems (ICITES)*, pages 131–137. IEEE, 2021.
- [100] Sebastian Rohjans, Klaus Piech, and Sebastian Lehnhoff. Uml-based modeling of opc ua address spaces for power systems. In *2013 IEEE International Workshop on Intelligent Energy Systems (IWIES)*, pages 209–214. IEEE, 2013.
- [101] Byunghun Lee, Dae-Kyoo Kim, Hyosik Yang, and Sungsoo Oh. Model transformation between opc ua and uml. *Computer Standards & Interfaces*, 50:236–250, 2017.

- [102] Florian Pauker, Sabine Wolny, Solmaz Mansour Fallah, and Manuel Wimmer. Uml2opc-  
uatransforming uml class diagrams to opc ua information models. *Procedia CIRP*, 67:128–133,  
2018.
- [103] Inc MagicLeap2. Download magic leap 2 developer tools, 2024.