TAMPERE UNIVERSITY OF TECHNOLOGY

OSCAR JESÚS NEIRA MILLÁN

MOBILE AUGMENTED REALITY APPLICATION FOR MONITORING INDUSTRIAL SYSTEMS

Master of Science Thesis

Examiner: Professor José L. M. Lastra
Examiners and topic approved in the
Automation, Mechanical and
Materials Engineering Faculty
Council meeting on 08th of May 2013

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Master of Science Degree Programme in Machine Automation
NEIRA MILLÁN, OSCAR JESÚS : Mobile Augmented Reality Application for Monitoring Industrial Systems
Master of Science Thesis, 94 pages, 16 Appendix pages
May 2013
Major: Factory Automation
Examiner: Prof. José Luis Martínez Lastra
Keywords: Virtual Reality, Augmented Reality, Mobile Augmented Reality, Monitoring, Mobile Devices, Wearable Devices, Manufacturing Systems, Factory Automation

Augmented Reality (AR) applications have constantly evolved during the past few years in the mobile technology field. The potential of Mobile Augmented Reality (MAR) permits to enrich the reality with digital information by integrating the physical context of the user, localization-based data, mobile embedded sensors and Internet connectivity makes out of MAR a promising technology to be used in a wide range of real use cases. Nowadays the AR communities are growing and more companies are investing in Virtual Reality (VR) and MAR technologies. Lots of applications have been developed for medical, military, entertainment, manufacturing and industrial environments. Besides this, with the exponential grow up of the mobile market, this type of applications are becoming closer to final users.

This work not only details the fundaments and concepts of AR but also describes the design and implementation of a MAR application within the area of industrial systems, specifically for the domain of discrete manufacturing. The application is designed to be used in mobile devices and special attention was taken, by using the latest technological trends in this area. The developed MAR application is intended to be used by people working with Manufacturing Systems (MS), allowing an intuitive and better visualization of huge amounts of data which are generated in the industrial system. An industrial testbed was used for testing this diploma work. Several tracking patterns were deployed along the production line to create points where the user can experience an AR interaction. The MAR application displays 3D objects in front of the user view by tracking the AR markers using different visualization gadgets, mobile devices or smart wearable devices (AR glasses).

The developed application is designed based on the requirements and specifications of a project named: Pro-active decision support for data-intensive environments (ASTUTE), supported by the European Union (EU). The final results of the implemented application is included as a part of ASTUTE, in the demonstrator of virtual control room for building and manufacturing process management.

PREFACE

The work described in this document represents more than the end of my Master degree, means the effort and the experience earned during the past two years working at Factory Automation Systems and Technologies Laboratory (FAST) in the Department of Production Engineering at Tampere University of Technology.

Firstly, I would like to thanks to Prof. Lastra for the opportunity and support working in FAST lab also for the supervision of my research plans. I have learned many things from his supervision and advice during this time.

I would like to thanks to Angelica Nieto for his advice and comments as my supervisor and friend, helping me to enhance my professional skills every day. Also, to the entire team in FAST laboratory. In particular, the team members working in ASTUTE project.

I want to thanks the persons who encouraged me to finish this work at the first moment, Fernando for giving me the first opportunity to come to Finland, to Axel for putting all his trust since the beginning as a coworker and friend, to my closest friends during this journey, Borja and Gerardo that always where there supporting me and making more pleasant the days working and studying, to my friends in Finland that cheer me up at any moment, teaching me that life is more than studying and working. At last but not least, to my friend Keops that even from the distance is always supporting me over the years.

Por ultimo quisiera agradecer a mi familia por el apoyo y las oportunidades que me han brindado, a mis padres por la educación y todas sus enseñanzas que me dieron la fuerza para seguir adelante, a mis hermanas que por su consejos y soporte incondicional en todos los proyectos de mi vida, a mi ahijado que fue mi inspiración en todo momento para terminar este trabajo.

Oscar Neira
Tampere, May 2013

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

AR	Augmented Reality
ADT	Android Developer Tools
ANDK	Android Native Developer Kit
AOS	Android Operating Systems
ASTUTE	Pro-active decision support for data-intensive environments
ARToolKit	Computer Tracking Library of Augmented Reality
AVITA	Augmented VIsualization for Transparent Factory
API	Application programming Interface
Android	Linux-based Operation Systems for Mobile Devices
Blender	Open-Source 3D Computer Graphics Software
BDD	Block Definition Diagram
CATIA	Computer Aided Three-dimentional Interactive Application
CAD	Computer-aided design
FA	Factory Automation
GPS	Global Positioning System
HMI	Human Machine Interfaces
HMD	Head Mounted Display
HDMI	High-Definition Multimedia Interface
HD	High-Definition
iOS	Apple Mobile Operating System
IDE	Integrated Development Environment
IR	Infrared Light
IT	Information Technologies
KINECT	MotionSensing Input Device

KPI	Key Performance Indicator
Linux	Unix Operating Systems
LHS	Laser Head Set
MAR	Mobile Augmented Reality
MS	Manufacturing Systems
MMS	Monitoring Manufacturing Systems
MacOS	Macintosh Operating Systems
MEMS	Micro-Electro-Mechanical Systems
NFC	Near Field Communication
OS	Operating Systems
OEE	Overall Equipment Effectiveness
RFID	Radio-Frequency Identification
SDK	Software Development Kit
SysML	System Modeling Language
UNITY3D	Cross-Platform Game Engine
USB	Universal Serial Bus
UML	Unified Modeling Language
VR	Virtual Reality
VAR	Vuforia Augmented Reality
WOS	Windows Operating Systems
WB	Web Browser

1. INTRODUCTION

The objective of this chapter is to describe the base of this thesis work. Section 1.1 is dedicated to the background of MAR and its applications. Based on the background, Section 1.2 presents the problem and justification needed to solve it. Section 1.3 establishes the objectives and methodology used along this document while Section 1.4 outlines the thesis.

1.1 Background

The emergence of Augmented Reality (AR) occurs after the implementation and development of the concept of Virtual Reality (VR). VR is a technology that immerses the user in a computer-generated world whereas AR combines the real world with computer graphics, mixing the physical reality with the virtual content [Mad11]. On the other hand, the term of AR was introduced in 1992 by Caudell and Miizell implementing an AR application overlaying computer-presented material on top of the real world, with the objective of helping aircraft assembly tasks [CM92]. The AR concept is led in different layers containing the attributes and functionalities that characterized AR technologies. These layers are: the Capture, the Register and the Augment, which represent the general principle of AR [BKLP04]. The Capture layer is the beginning of any AR application and input signal from the real world. Afterwards, the Register layer is in charge of the alignment of the virtual representation to the real world. Finally, the Augment layer creates an augmentation of the virtual objects by replacing the real with the virtual objects.

The description of MAR begins to be used and referring due to the fact that MAR is a special instance of AR. So far, the technologies behind MAR are basically the same in comparison with AR. Currently, these applications reached great expectations due to the exponentially growing of the market in mobile devices, accordingly to the main concept of MAR. The main functionalities of MAR applications are supported by the mobility of this technology, used for localization and positioning of the device when the application is running, in fact, the mobility in MAR applications is considered as the principal advantage. As a result, due to the changes in manufacturing, marketing and development, new hardware and software emerge making MAR a wide open solution, allowing the creation of new devices and applications based on MAR.

In 1999, the first demonstration arrived at the market a Computer Tracking Library of Augmented Reality (ARToolKit) displaying basic 3D elements over images (patterns) now known as AR markers, also giving support for different operating systems including open source license for free use [Kat99]. Then, after some years of development with AR many other applications appears to integrate AR using additional technologies. For instance, the Web Browser (WB) is now a clear example of technology adopting AR as an option to display information to the user. However, the WB has also been changed during the past few years. Now a WB is a huge influence in the software market also used as an instance of AR [vdKBLF09, Wik08, Jun09]. The value of the mentioned integration of AR technologies into a WB is the capability of data fusion based on information taken from the real environment, for instance, the mobile devices with information taken from the user localization-based data (Global Positioning System (GPS)) and embedded sensors including the compass, gyroscope and accelerometers. The data fusion is processed through the application in order to have a precise information about the positioning of the device. Meanwhile, many other applications have been developed by different approaches, some creating from scratch their own solution of AR [Qua12, Met03] and others still including ARToolKit as a base of their system [ART04].

Currently, the trend of technology is changing to the use of wearable devices in combination with other developments such as cloud computing, Internet of things and mobile phones. This tendency causes the difference in the actual IT market modifying the scope of investment. For that reason, companies like Apple, Google, Microsoft and others, have begun the competition in this domain, creating better hardware and software applications including new features merging design, accuracy and social aspects. In particular, the actual status of AR is divided by different the companies that produce the components, and others that implement the platforms strongly influenced for mobile devices and wearable devices.

1.2 Problem Definition

Nowadays, visualization systems are everywhere and AR it is not the exception. The current research found AR as a potential technology trend, showing an exponential applied growth in different domains. For this reason, the manufacturing domain has been influenced by the development in AR, creating new applications to control and monitor the production lines. Until now, the applications used with AR in manufacturing systems only show basic information to accomplish simple tasks of maintenance, performance, issues, malfunctions, process and operations.

On the other hand, the industrial domain still is developing new applications including many technologies like touch screen panels or personal computers, that can also be replaced with other solutions as is the migration to mobile devices (smartphones and tablets) providing different perspectives when the user is working in the system and suddenly an event occurs. The use of mobile devices is becoming a huge opportunity to develop new applications in the domain of Factory Automation (FA). This research aims to propose and integrate a new development approach in MAR for monitoring industrial systems, merging new trends of technologies with mobile devices and wearable devices improving the interaction between the users and the manufacturing systems.

1.2.1 Justification of the Work

In the past few years, many technologies have emerged using VR, due to the fast growing of technologies in different domains, for example, entertainment, military, medical and industrial. For this reason, new trends of technologies are starting to lead the IT market and mobile market. In fact, AR is one of those potential trends for the future integration with smart devices and wearable devices and not only for a small market this idea covers different purposes and domains where AR can be applied, starting for the reality nowadays where most of the AR applications are developed for mobile devices.

On the other hand, MAR emerges due to the exponential growth in the mobile market. In addition, to the fact that millions of mobile devices are sold every day around the world and the increase of investment in this technology due to the large number of applications developed for these platforms, for this reason, MAR was considered as an instance of AR. In conclusion, the AR market is wide open and is reaching great potential to be improved in different ways. Meanwhile, Manufacturing Systems (MS) have improved at the same time with new trends of technologies related with VR and AR including technologies such as mobile, smart, embedded and wearable devices. As a result, new applications emerge giving more and new solutions for MS. For example, a control application that improves the performance of the factory floor, retrieving and delivering the current information around the production line. However, some of the proposed solutions are statics and others are integrated with more intelligent systems capable to create or generate adaptable solutions to the users in MS [NLLC13].

The implementation of AR can be useful to complete daily tasks in the factory floor saving time and resources, sending and retrieving information based on the position and the actual status of the user. These applications can solve some issues related with the factory automation domain such as the mobility, positioning and acquisition of information based on the context of the user.

The mobility will be one of the main advantages inside the factory floor and will save time using only one device to complete many tasks allowing also to send reports of alarms of the current events in the production line. In addition, the proposed solution will make more comfortable the use of monitoring and controlling systems due the handy use of mobile and wearable in real scenarios for MS. In conclusion, taking the results of the process of development and implementation the application can be adapted for different simple or complex scenarios inside the factory floor.

1.2.2 Problem Statement

As previously explained, there are many types of applications in MS using VR and AR, that integrate features such as localization-based data and embedded sensors. Thus, the adaptation of these technologies in a complex scenario becomes a difficult task to be faced by developers. Note that most of these applications are used with simple content designs without integration to other modules in the system. As a result, there is a wide opportunity to design and develop new applications based on the complexity of the MS and the constant increase of the industrial or mobile market. For this reason, it is required to integrate more intelligent systems merging industrial systems and more powerful graphic software to cover the lack of design and performance. The MAR should be very specific using real scenarios based on industrial environments or current manufacturing systems, allowing the users to have more precise visual information to solve new problems in the factory. In addition, the system should be able to establish communication with the actual manufacturing process at all levels. These statements prompt the following questions:

- How to integrate different technologies to create a AR application?
- How to generate a MAR application capable of deliver different experiences to the user showing better quality visual information about the current state of the system, the layout of the industrial equipment, the production and events around the user in a MS?
- How to provide a mechanism to establish a connection between low level layers and high layers?
- Hence, how to integrate the MS thought industrial testbed scenarios with a MAR application?

1.3 Work Description

1.3.1 Objectives

The main objective of this research work is the creation of a solution based on AR for visualization and integration with Monitoring Manufacturing Systems (MMS) including technologies as mobile and wearable devices like Vuzix Smart Glasses M100 and Google Glasses. These smart devices would enhance the perspective of the reality overlapping virtual objects on images from the real world. Finally, the MAR application should merge AR technologies with smartphones, tablet and smart glasses to the proposed solution in order to deliver a visual monitoring application for industrial use cases. In other words, the MAR application must start overlapping 3D objects over images generated by the camera in a mobile device or smart glasses, that display 3D content over images generated by the human eye. The following list describes the objectives proposed during this thesis work.

1. Thoroughly planning and analyze existing AR applications in the domain of MMS with positive and negative aspects in their design and implementation.

2. Planning and Analysis the MAR application based on the testbed description and current research.

3. Design of a MAR application based on results of the sequential methodology improving the processes containing elements like requirement diagrams, use case diagrams, sequential diagrams and block definition diagrams to define and illustrate the description of the testbed.

4. Create an AR application integrating their basic elements, deploying the results in mobile devices and wearable devices, making different adaptations for mobile devices including smartphones, tablet and smart glasses.

5. Establish a connection with other modules of the MMS, retrieving and sending information through the modules of the system, showing the current state of the system at runtime.

6. Implementation of a dynamic and attractive MAR application used for testing purposes. Based on results of the usability test the data will be analyzes and compiled for future work.

7. Create a stable MAR version based on the usability test results including previous feedbacks and comments given by the team of project ASTUTE.

1.3.2 Methodology

Literature review and analysis of the state-of-the-art related with Virtual Reality (VR), Augmented Reality (AR) and Mobile Augmented Reality (MAR).

The most important trends of technologies related VR are studied and reviewed from the basic concept of VR. In addition, this review describes the evolution of VR including references of AR and their instances.

The VR system structure is described with more details and their elements are introduced by sections including inputs, outputs, engine, software and databases. In addition, each subsection shows and analyzes the new trends of technologies applied in different domains. Finally, some examples of virtual programming environments are described in order to conclude with the process of development of VR systems.

The history and elements of AR are also studied and reviewed from the first application prototype based on AR until the first AR web browser application. As a result, a detailed list of solutions are described within different AR libraries options that currently are used in the market.

Application of software development methodology to integrate a MAR solution. Based on a sequential development process to generate requirements, visual diagrams, designs of user interfaces and workflows.

Based on the initial testbed description the functional and nonfunctional requirements should be identified. As a result, the requirements are integrated into tables generating SysML diagrams, in order to create initial interactions between the MAR application and the users.

Furthermore, the design phase incorporates sequential diagrams in order to describe the data flow between the modules described in previous processes. As a result, the user interface is designed. In addition, other components of the system are defined such as the elements of the interface, the AR marker, type of users, type of devices and user interactions.

Finally, a general block definition diagram is generated based on the entire information flow, describing the elements in a general view. The blocks represent the entire solution of integration of a MAR application. The block definition diagram is also improved accordingly in the interaction of the selected sequential methodology.

Note that the selection of the AR library is essential to achieve the objectives of this work. The selected library needs to fulfill different requirements to be integrated into a mobile device also in a suitable programming environment to establish communication with the generated virtual objects.

The selection of a programming environment is also a main priority, the selected Integrated Development Environment (IDE) must be able to import the libraries of AR and it should be capable of establishing communication between the created 3D objects in the scene and the algorithms that contain the layers of AR.

Implementation of the Mobile Augmented Reality (MAR) application in Manufacturing Systems (MS)

The list of selected hardware for testing is provided in this section in order to explain the installation and the configuration of the selected hardware, including additional software such as the libraries of external sources, Software Development Kit (SDK) and configuration manuals. Based on the system design, the MAR application is integrated within the list of technologies mentioned along this work, including external systems which are described in the testbed.

Empirical study

The application is developed for mobile devices and wearable devices. The MAR application in their initial state is integrated specially for smartphones and tables including Android Operating Systems (AOS). In addition, the application contains few functionalities for monitoring manufacturing systems. After, several interactions with the sequential development methodology, a stable version is created to be tested in an usability test. On the other hand, empirical studies are performed over a manufacturing system in particular a production line that performs assemblies of components and is composed by workstations, robots, conveyors, buffers and pallets. Note that the initial testbed is formed by the previous cited production line elements. In addition, a third-party application simulates some behaviors of the production line in order to prove and display some result during the tests with the connection module. Finally, the user is free to interact with the application and see the responses of the MAR application. Then. the data is analyzed and compiled for future work and categorized by priority.

1.3.3 Assumptions and Limitations

The focus of the work is aimed to monitor the production in MS. The implementation of the MAR application is described as well the installation with other communication modules with external and internal systems. For this reason, some of these sources out of this document are described in the same way, in order to see the integration with the other modules.

Assumption 1: The production line is always on, with a continuous production of products, at the same time the pallets are following the desire flow of the production line in a controlled environment.

Assumption 2: The manufacturing system is always retrieving and sending data through the applications in the factory floor.

Assumption 3: The MAR application at the same time is always receiving and sending data.

Assumption 4: The connections with external information sources such as databases are always updated.

Assumption 5: The manufacturing is always generating current information given by the the factory floor.

Assumption 6: The users have experience with smartphones, tables and smart glasses.

Assumption 7: The users knows and have experience with VR, AR and MAR applications.

Assumption 8: The user is always using one of the proposed devices in this document including the corresponding version of the MAR application.

1.4 Thesis Outline

The thesis work is structured as follows; Chapter 2 presents an introduction of the used technologies, their state-of-the-art related VR, AR, MAR including instances and applications in different domains. Chapter 3 describes the sequential development methodology that is proposed for planning, analyze, design, implement and test the MAR application including some references to the testbed for performing the study. Chapter 4 contains the recipe to implement a MAR application. In addition, the description of the main functionalities of the application is added in this Chapter in order to explain the communication between the MAR application and the components of the users interface. Chapter 5 presents the results of the study based on the usability test. Finally, Chapter 6 presents the analysis of the obtained results, conclusion and further work in the area of VR, focusing on the development of MAR applications.

2. LITERATURE AND TECHNOLOGY REVIEW

2.1 Virtual Reality

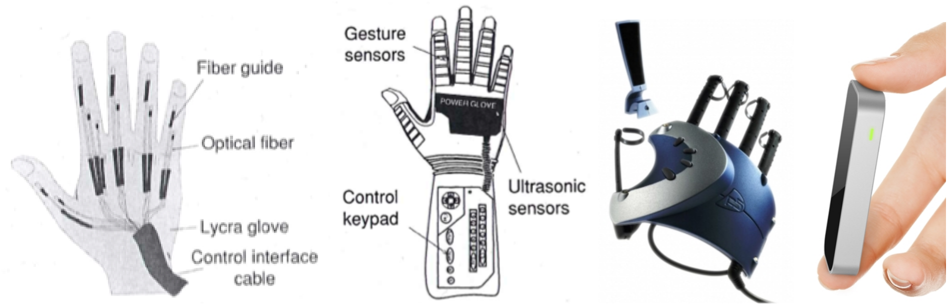
The scientific community has been working in the field of VR for decades and nowadays there are a large number of publications and applications related VR and their instances such as AR and MAR. Moreover, there are still some confusions even in the technical literature about what it is VR. In terms of functionality, VR can be seen as a simulation in which computer graphics is used to create a realistic-looking world [BC03]. However, the definition can vary based on the time where is defined and implemented, for instance, Steve Bryson (NASA Ames) define VR as the use of computer technology to create the effect of an interactive three-dimensional world in which the objects have a sense of spatial presence [Bry]. On the contrary, seeing this from another angle, John Briggs define VR as a three dimensional, computer generated simulation in which can navigate around, interact with, and be immersed in another environment [Bri96]. In addition, other references introduce a proper definition of VR as a high-end user-computer interface that involves real time simulation and interaction trough multiple sensorial channels. These sensorial modalities are visual, auditory, tactile, smell, and taste [BC03].

Thus, the definition has been changed due to the new trends of technology involving VR an their instances such as AR and MAR defining VR as a complementary artificial digital environment that uses computer hardware and software to create the appearance of the real environment to the user [KR13]. Nowadays, VR definition is not needed anymore, due to the market influence in movies, commercials, publicity campaigns and applications that involves the user in a complete or partial virtual reality.

The evolution of VR lies in more that 40 years of references including articles and books, that describe the concept of the immersion to a virtual environment. An idea started in 1962 U.S. issued to Morton Heilig for his invention entitled Sensorama Simulator, which was the first VR video arcade [BC03]. Further on, due the actual exponential change in different domains applying VR technologies other technologies rises also as instances derived from VR. In fact, adopted from fields like entertainment, medical and military. For example, the TV series and movies showing their perspective of what is the immersion of the user in virtual environment, ideas that are more close everyday and in some cases a reality.

In the mean time, after the first tests of VR, other developments show up in the research field, integrating new interfaces to manipulate the virtual environment such as the sensing gloves (see Figure 2.1(a)) was the first commercial VR technology integrating a new interface in comparison with the standard interfaces of the time (and still today) the keyboard and the mouse. On the other hand, The VPL DataGlove introduces gestures to the system based on the measuring of the finger and thump bending using fiber-optics sensors. After, the release of the VPL DataGlove companies started the development of cheaper technologies to be introduced in the market, for instance, Nintendo creating new alternatives to interact with the virtual environment as a derivation of sensing gloves called PowerGlove (see Figure 2.1(b)) by using ultrasonic sensors to measure the wrist position relative to the position of the screen and conductive ink flex sensors to measure the finger bending.

The sales of the Power Glove exceeded the expectation of the market, but as the history defines this device and the others no mentioned in this text after the release of another innovation product integrating better design and better functionalities, the previous device is considered as a piece of history, where one day everybody will remember it as an old fashion tool. As an illustration, the PowerGlove due the lack of games, other platforms start to pop up such as the beginning of the console games era (Sony Play Station). Thus, the history of VR can be reflected in the given examples taking the sensing gloves, now with technologies as is the P5Glove (see Figure 2.1(c)) also share with the previous devices some features, the main deference lies in the design and accuracy integrating tracking system with 6 degrees of freedom. Finally, the need of using external devices (wearable devices) to communicate and send operation to the machine is now written in books of the history of VR with trends of technology using optical trackers like Infrared Light (IR) cameras capturing the gestures of the user creating an interaction with the VR system like Microsoft with MotionSensing Input Device (KINECT).



(a) (b) (c) (d)

Figure 2.1: Sensing Glove technologies, a comparison with new trends of technologies: (a) the VPL DataGlove [BC03]. (b) the PowerGlove [BC03]. (c) Essential Reality P5 Glove. (d) LEAP motion controller [Mot12].

As a matter of fact, due the new necessities of the users, new requirements emerge and now companies started filling this gap developing new devices even faster and smaller, for example, the Leap motion controller (Figure 2.1(d)) designed to create an easy interaction between the human and the machines, using it as an input device to the computer, taking into account previous examples in Section 2.1. The Leap motion controller is a small Universal Serial Bus (USB) peripheral device designed to be placed on a physical desktop, facing upwards, using the IR cameras to observe a roughly hemispherical area to track the movements of the user, designed to track small items such as a pen and chopsticks. The Figure 2.2 shows how the user interacting with the computer based on the Leap motion controller.

Figure 2.2: LEAP Motion Controller: User-Machine Interaction.

VR is an industry, and is reflected by the actual market and is still involved in new trends of technologies derived from VR. Technologies with VR are overwhelming the market even more with the tendency of mobile devices to create a big network of users and applications based on virtual environments nowadays called MAR. Moreover, these applications are not the only emerging innovation, technologies defined as wearable devices are returning as new trends of smart devices with huge expectatives to be implemented in different domains like entertainment, social and work (Project Glass [Gog12]).

2.1.1 Structure of Virtual Reality Systems

VR is composed by many elements some of them mentioned in previous sections 2.1. Now, is possible to describe VR with more details integrating more elements between the architecture and the concept of VR connecting the real world using inputs and outputs. The following Figure 2.3 shows the general elements of a VR system and their interaction with other devices and systems.

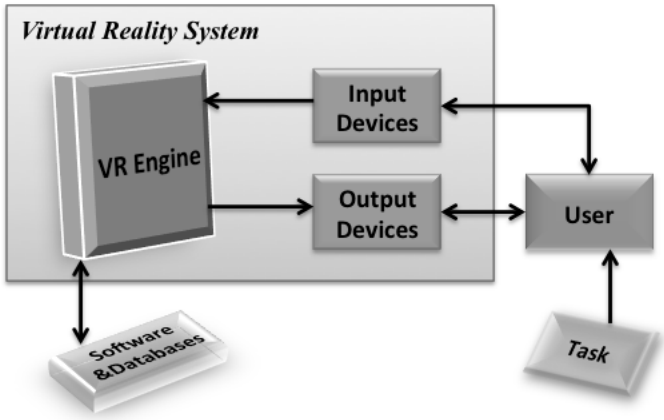


Figure 2.3: Virtual Reality (VR) System Architecture [BC03].

In order to allow the interaction between the human-computer it is needed the use of special interfaces to send information to the VR engine. Then, after the inputs have reached the VR engine. Moreover, after the VR engine processes that information, the VR engine starts providing feedback for the output devices in different forms such as sounds, haptics or images. Thus, input and output devices merge the integration of Virtual Reality (VR). The following subsections analyze some the input and output devices describing how the interaction is created by the VR engine.

In different ways the communication between human-computer can be defined, an interface is a means of communication between users (or users) and system. The user communicates commands, request, questions, intents and goals. The system in turn must provide feedback for those actions sent by the user. For example, a request input searching for information related with the system state [BKLP04]. However, the user and the system do not speak the same language and they required a series of translations to establish a communication. Thus, that is why the interface is the media that serve as an intermediary translator between the user and the VR system. In addition, there are multiple translations steps involved in a human-computer interaction.

In other words, Figure 2.4 describes those interfaces starting from the user and the action performed. Then, the input device transform those physical actions to electronic signals to the system. Finally, the system deciphers the signal from the input device and delivers based on the actual state of the system more information (System goals) to the output device. In the end, the information is displayed by the output device, information that the user can perceive such as the light and sound.

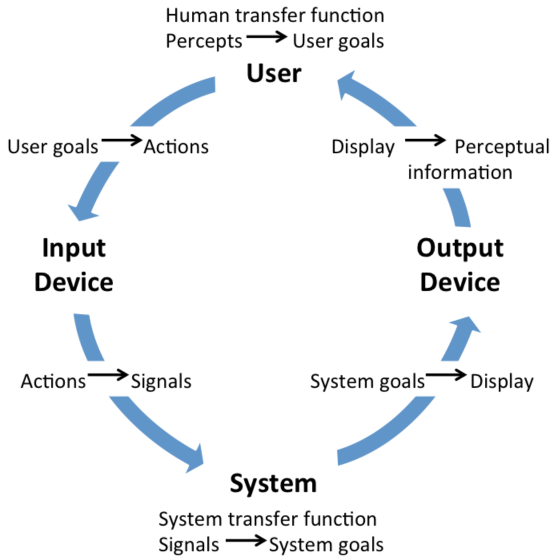


Figure 2.4: Human-computer Communication Through Devices [BKLP04].

2.1.2 Virtual Reality Input Devices

The Virtual Reality (VR) inputs are devices that interact with the system capturing gestures from the user. The device itself transforms these movements, sending the information to the VR system. Historically, there are different types of inputs devices some currently used in research fields others used for commercial purposes.

Three-Dimensional Position Trackers

Many computer application domains such as navigation, ballistic missile tracking, ubiquitous computing, robotics, biomechanics, architecture, Computer-aided design (CAD), education, and VR, require knowledge of the real-time position and orientation of the moving objects within the frame of reference [BC03]. These applications are different and vary depending on the requirements of the application and some of them need more precision and accuracy than others, in the same way they work with the same principle. For instance, using the example of a 3D objects moving along of a 3D space.

These objects have six degrees of freedom, three for translations and three for rotations. As is illustrated in the following Figure 2.5, if a Cartesian coordinate system is attached to the moving object, the translations refer along the X,Y and Z and the rotations about the axes are rolled yaw, pitch and roll, respectively. The data taken from this model need to be measured sufficiently rapidly, as the object is moving. For definition, a special-purpose hardware used in VR to measure the real-time change in a 3D object position and orientation is called a tracker [BC03].

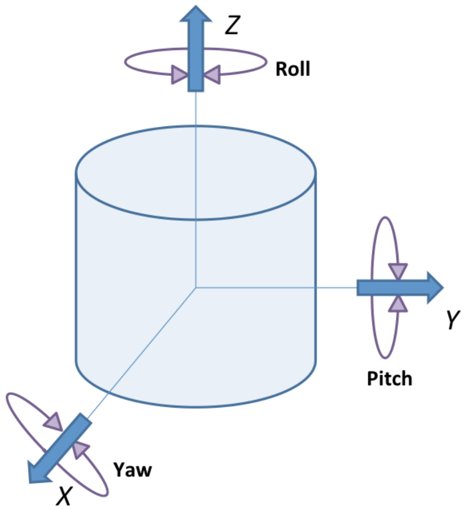


Figure 2.5: VR System of Coordinates, Moving 3D objects [BC03].

The Three-dimensional position trackers are divided into different categories starting from the first VR trackers. For instance, the Mechanical Trackers, Magnetic Trackers, Ultrasonic Trackers, Optical Trackers, Hybrid Inertial Trackers. Due to the purpose of this section, some elements of VR trackers are mentioned and described with more details to understand the meaning of the different types of the three-dimensional position trackers. After analyze the different types of tracker also is taken into consideration the performance of each type of elements, as a result of the principle of each of them.

The performance of the tracker is one of the most important element needed in order to see if the selected tracker covers the requirements of the VR system. In addition, the accuracy of the tracker in definition can be seen as the representation of the difference between the object's actual 3D position and the report by the tracker measures [BC03]. In the end, to achieve a better tracker performance it is also needed the integration of other elements merging the accuracy and performance such as the jitter, drift and latency.

Further on, going back to the new trends of technologies related with the main subject of this section (input devices for VR systems), only some cases are taken into this section, as an example of input devices defined as Three-dimensional position tracker, continuing with information related with the general subject of this document.

The optical trackers are becoming more attractive due to the low cost and high resolution, additionally to the fact that can be used as an input device in most of the VR systems. Currently, some applications in different domains are referencing some implemented technologies in this document such as the tracking algorithms in real-time. As a result, in comparison with other applications the 3D low-cost optical position trackers are analyzed with more details, starting from the definition of the technology and their elements where are involved. In addition, some scientific documents are described including the demonstration of this technology and their applications.

The optical tracker is defined as a non contact position measurement device that uses optical sensing to determine the real-time position/orientation of an object [BC03]. The most famous example is a technology developed in 2010 by Microsoft (KINECT), some implementations have been made integrating this device in systems to track objects in real-time, for example, using Simulink, in the example given in [FYJ10] the selection of the hardware is described in Figure 2.6 also other examples such as [NDI+11] and [FP11] using the same equipment KINECT for hand tracking and rendering in wearable haptic devices, making a KINECT Fusion for real-time dense surface mapping and tracking. These applications have common functionalities, as is the acquisition of images from the real world based on the IR cameras. However, the difference lies in how they process that information to their VR system.

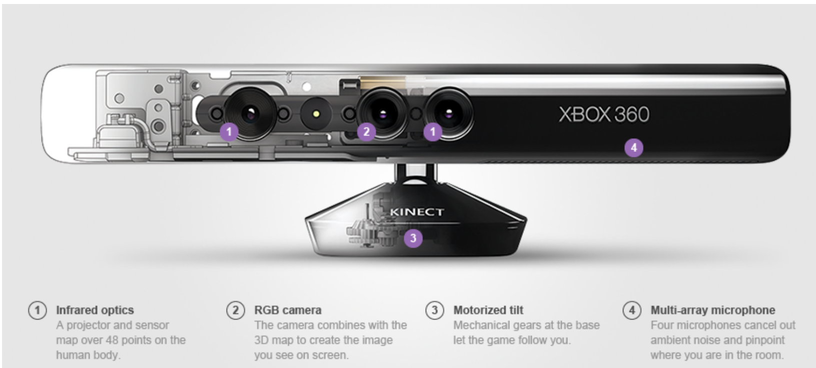


Figure 2.6: Element Description MotionSensing Input Device (KINECT).

As a result, this device can be used for everyone, is cheap and can be applied for different purposes, the most basic common element is the tracking and mapping. As is illustrated in Figure 2.7(a) a image that can be proceeded by the VR system, the image is taken from the RGB camera and is useful to be compared with the image Figure 2.7(b) where the image is taken from the IR camera and ready to be analyzed by the VR system, in this example the images are acquired and analyzed in Simulink to track objects in real-time. The pictures are taken by a standard KINECT device and implemented in a research field [FYJ10].

(a) KINECT RGB camera. (b) KINECT Infrared Light (IR) camera.

Figure 2.7: Images Taken from Microsoft MotionSensing Input Device (KINECT). (a) KINECT RGB camera [FYJ10]. (b) KINECT Infrared Light (IR) camera [FYJ10].

Gesture Interfaces

A gesture interface is defined as a device that measure the real-time position of the user's finger as is mentioned in the history of VR systems (see Figure 2.1), in order to allow natural, gesture-recognition based interaction with virtual environments. Sensing Gloves (Figure 2.1) are the best known gesture interface in the market, these devices have embedded sensors which measure the position of each finger versus the palm.

There are different types of Sensing Gloves and differ from the type of sensors that they use, the number of sensors in each finger and glove and in the resolution of type of connectivity (wireless, bluetooth, Infrared Light (IR)). So far, there are different types of Sensing Gloves devices, for instance, the Pinch Glove was one of the first devices in the market acquiring gestures from the user, after some testing one of the drawbacks of this technology was the calibration of the device each time the gloves change of user. However, as is mentioned in Section 2.1 these devices are deprecated at some point. As a result, new devices emerge into the market with great proposals such as the LEAP motion controller, Figure 2.8 shows the controller seeing from a technical perspective, describing the elements inside this device. The LEAP controller contains two cameras that allow the acquisition of different objects including finger, the hand even a pen can be detected in order to have more precision in more advance applications.

Figure 2.8: LEAP Motion Controller Device.

2.1.3 Virtual Reality Output Devices

The output devices in Virtual Reality (VR) systems are designed to provide feedback in response to the action taken by the user or generate data based on calculations from the VR engine. The sensorial channels are taken into account by these interfaces. These channels are included by categories including the sight (graphics displays), sounds (3D sound displays), and touch (haptic displays). The section describes some of the common devices on the market, taking into consideration the graphics displays mentioned along this document.

Visual displays present information to the user through the human visual system and can be considered as a 3D user interface [BKLP04]. In addition the visual displays have a number of important characteristics from the 3D interactive perspective, in this subject it is needed to considered some important concept such as the field of regard, field of view, spatial resolution, screen geometry, light transfer mechanism, refresh rate and ergonomics. A graphic display is characterized according to the type of the image produced (stereoscopic or monochromatic), their image resolution (number of pixels in the scene), the field of view (field of vision) and display technology(LCD- or CTR-based).

The graphic displays use the more powerful human sensor channel, with an extremely large processing bandwidth, some of the VR systems may not incorporate sound or haptic feedback, but all will have some type of graphic display. In addition, the graphic displays can be categorized by personal graphic displays which is destined to be viewed by a single user [BC03]. Thus, between the personal graphic display there are categories and are divided as HMDs, hand-supported displays, floor-supported displays, and autostereoscopic monitors.

As is mentioned along this document the technologies with more common functionalities to create VR systems, precisely the instance of VR such as AR and MAR are taken as a priority.

Thus, in this section will be taken into consideration different categories of graphic displays as HMD, describing the basic principle of this technology, as well the present research related with these devices like Google with Project Glass [Gog12] and Vuzix with Smart Glasses M100 [Vuz12]. The HMD projects an image in front of the user this can change depending on the device and the technology as is discussed in previous sections referring the historical background of VR systems. So far, in this scenario the old HMD devices have very low resolution and incorporate a diffuser sheet overlaid on the input to their optics. Modern HMDs have resolutions up to extend video graphics array (XVGA; 1024 x 768) [BC03]. Moreover, the characteristics of HMD such as the display technology, weight, comfort and cost are additional criteria to be considered in the implementation and development of these devices.

The principle of HMD is that use two small displays embedded in a helmet, using head tracking improving the immersion. Then, the user can look around by moving its head. As an Illustration, the following Figure 2.9 describes a high resolution AMLCD displays located in the user forehead, each display has three separate LCD panels, the price on the ProView XL35 comes at a price of terms of much higher costs and weight compared with other technologies.

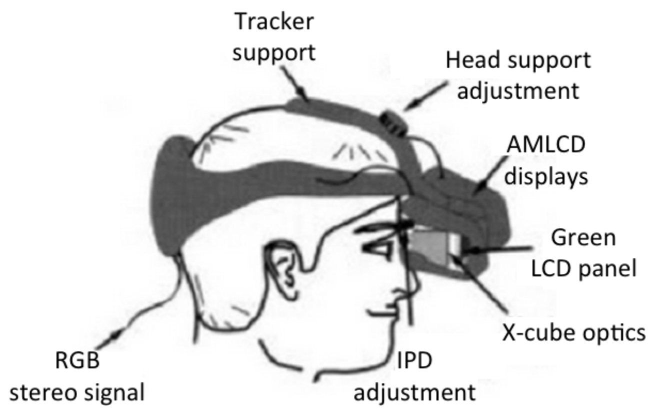
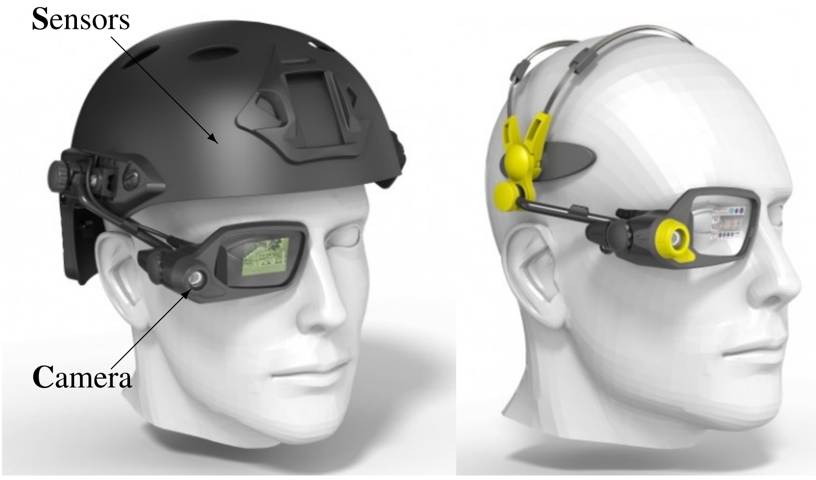


Figure 2.9: Stereoscopic HMDs, ProViewXL35. Adapted from KaiserElectro-Optics[2001].

In the last decade, the research related graphic displays increased, due to the necessity of the user to acquire these devices for making their life more comfortable in home, work even traveling. Vuzix Corporation is a company formed in 1997 manufacturing and sells computer displays devices and software. For example, Figure 2.10(a) shows a prototype product, presented by Vuzix in a press release on CES2012. The Smart Helmet device integrates a High Resolution display and can be used in workstations for industrial environments or military applications.

The Smart Helmet covers only a portion of the market that the company wants, but that does not mean that the device cannot be improved and integrated with more functionalities. However, this device for a normal customer can be an unnecessary device only with the simple idea of wearing a helmet all day long. Figure 2.10(b) shows another version of the product development in Vuzix, this device integrates the same capabilities of the Smart Helmet also with brightness and contrast good enough for outside use. The output pass through a 1.4 mm thick plastic waveguide lens and the image is extended in 2D into the user's eyes.



(a) Vuzix Corporation Smart Helmet press release prototype

(b) Vuzix Corporation Smart Glasses prototype.

Figure 2.10: Images from press release on Consumer Electronics Show 2012 [Cor12a]. (a) Smart Helmet prototype. (b) Smart Glasses device prototype.

Vuzix M2000AR is the first Augmented Reality (AR) system designed specially for end-user use in industrial applications. This rugged monocular AR system is designed for use in the field, as an example, Figure 2.11(a) shows the components of the first AR industrial device. The Vuzix M2000AR is constructed by analogized aluminum and its water proof for real work conditions. Also, contains a monochrome display and is the first use Vuzix SMART Glasses technology. The Vuzix M2000AR allows the maintenance to training personnel to access important information without interrupting their view of the world around them (Figure 2.11(b)). Adding a hi-resolution camera and motion sensors, built into aluminum alloys housings, makes the M2000AR the latest fully AR capable personal display system in a tough industrial duty package. The two models, green monochrome and full color, are available and can be connected to an assortment of High-Definition Multimedia Interface (HDMI) and USB equipped host devices, including mobile phones, tablets, portable laptop and desktop computers.

(a) Vuzix M2000AR

(b) Vuzix M2000AR in a real scenario.

Figure 2.11: Images taken from Vuzix Corporation web page, product's brochure [Cor12b]. (a) Vuzix M2000AR image prototype. (b) Vuzix M2000AR device in real scenario.

The usage of this device includes the Hands-free viewing of technical data and full color multi-media content provided by a variety of host devices, the viewing video or instrument displays without a dangerous loss of spatial awareness, access AR systems for interactive assistance with maintenance and repair procedures and record and playback maintenance and repair procedures. Figure 2.12 shows the main element of the Vuzix M2000AR. The display of this device has a resolution of 1280 x 720 (720p) HD colors with a field of view of 30 degrees (diagonal), this is equivalent to 13" laptop at 24" or 64". In addition, it is mentioned in the data sheet description, the device is suitable for outdoor use with >400 nits-full color and >1200 nits-monochrome green including a HDMI 1.2 video interface with 2-3 hour battery life, rechargeable lithium power back and a connection for auxiliary power back. Finally, this single piece includes a 5 megapixel camera with high speed (up to 83 fps, VGA) video capture.

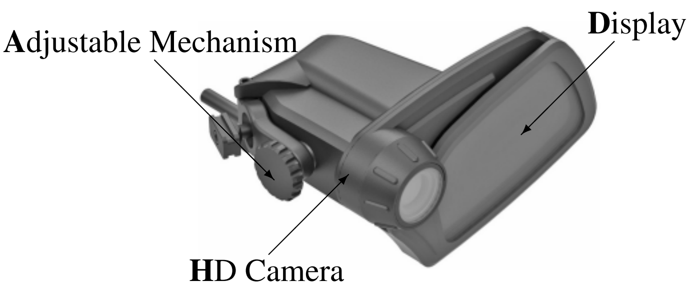


Figure 2.12: Vuzix M2000AR Smart Glasses Main Components [Cor12b].

On another hand, the wearable technology is evolving to target the consumer market. Now wearable computers are likely to replace smartphones and enhance communication and technology in daily life. However, today there is a great expectation of these devices to show up. The first example is the laser projector technology, developed by Fijitsu creating the Laser Head Set (LHS) device only available in exhibitions. The LHS is an audio-visual headphone, enable to create a fantastic audio-visual experience and is a prototype for future miniature equipment. This device was shown in Mobile Wold Congress 2013 with the objective to get more customers and investors. The technology delivers a wide viewing angle of 40 degrees with a large-size resolution view, also operated by PC or smartphone and could be a next generation IT eyewear. Unfortunately, this device is under development without a date of release and the prototype is not very comfortable. Finally, Figure 2.13(a) shows the headphones and the retractile screen. In contrast, Figure 2.13(b) shows the retractable screen in position.

(a) Laser Head Set (LHS) Headphones.

(b) Laser Head Set (LHS) Retractable Display.

Figure 2.13: Images taken from pamphlet given in Mobile World Congress 2013. (a) Laser Head Set (LHS) giving a perspective of the device [QDL13]. (b) Small screen adjustable in device Laser Head Set (LHS) [QDL13].

In comparison the LHS with the new era of mobile technologies is transforming the development and acquisition of HMDs, creating devices even more smaller, thin and more or less cheaper than before, making more interesting the competition related to the marketing and development strategics between the companies. So far, the companies as Briliant Service Corporation also present in Mobile World Congress 2013 started developing the same type of devices as is QDLaser [QDL13]. This device is more ergonomic and adjustable to the user.

The following Figure 2.14 shows the elements that integrate this prototype also some screenshots of the general idea of the device putting in practice real scenarios. The VIKING is an wearable device with some differences with the LHS [QDL13] and Goggle Glasses [Gog12]. The main difference is the position of the components such as the camera, sensor, GPS and battery, the only difference is the integration of two small screens. However, this device has no publications about the component inside the device or a deadline for of delivering developers kits.

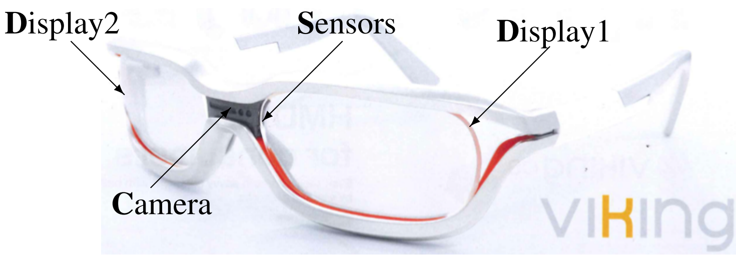
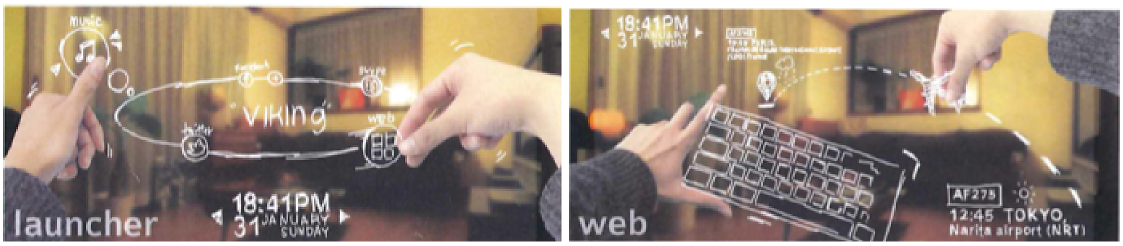


Figure 2.14: VIKING Smart Glasses Brilliant Service Corporation [Bri12].

The basic idea with VIKING smart glasses [Bri12] is the integration of AR in any activity of the user. The programming language used for the software side is Objective-C, this means that the basic interface development will be distributed for Macintosh Operating Systems (MacOS) especially for mobile devices with Apple Mobile Operating System (iOS). The Figure 2.15(a) shows the interaction with the user and the interface, in this example, the integration of a "Luncher" interface, after the user selects the WB application. Then, the information is displayed in the device, showing the relevant information to the user in that moment (see Figure 2.15(b)).



(a) Luncher application from VIKING. (b) Application after luncher a Web Browser (WB).

Figure 2.15: Images taken from pamphlet given in Mobile World Congress 2013. (a) View using VIKING Smart Glasses lunching an application. (b) Information displayed by the application after lunching the Web Browser (WB).

Meanwhile, in comparison with the last two prototypes mentioned, there is other option available for developers in the United States and Europe. In fact, will be a strong participant in the market of wearable devices against Project Glass [Gog12]. Vizux Corporation also has their own product and is a considerable option for competing with Project Glass. The Vuzix Smart Glasses M100 is the wold's first "Hand Free" smartphone display for on-the-go data access from and your Smartphone and internet.

The same principle of AR is applied for this device including for the rest of the product in the catalogue of AR systems from Vuzix Corporation. The next Figure 2.16 describes the concept application of the Smart Glasses M100. The integration of this technology includes hardware and software, ready to use for common applications such as the mail or social networks, containing a small screen with display resolution of WQVGA color 16x9 displays and High-Definition (HD) camera with 1080p to capture videos and images. The Vuzix Smart Glasses will be described with more details in Chapter 4 in order to describe the main component of this device and how was implemented in real scenarios for industrial environments.

Figure 2.16: Vuzix Smart Glasses M100: first "Hand Free" smartphone display for on-the-go data access from and your Smartphone and internet [Vuz12].

Finally, a project that chance the direction of the new trends of technology of wearable smart systems. Project Glass [Gog12] starts in 2012 with a press release in the annual conference for developers, Google I/O 2012. This development combines hardware and software, that delivers a different experience with AR including more functions integrating sounds, voice recognition, video calls, GPS and sensors added to deliver a better experience with AR and the real world. In addition, to the entire package of services that Google offers. For this and more reasons, this device is the most known in the market and only few developers have the opportunity to start developing in this device.

The relevance on this technology is demonstrated since articles, magazines and news around the world are describing the huge change that this technology can cause in the future. However, some of them start making their own solution and showing how to create your own smart glasses with cheaper elements (see[Fur13]), using the hardware from different sources of manufacturing vendors giving another options to create your own wearable device. Figure 2.17 shows the initial prototype of Project Glass including its components in the first stage of development. In contrast, the difference between the initial prototype to the prototype released in 2012 is obvious.

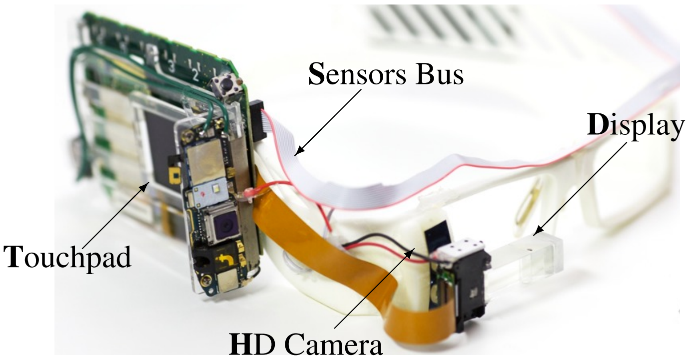


Figure 2.17: Project Glass Initial Prototype.

In comparison with the initial prototype of Google, the newest device shows a different frame, smaller than the first one also the components are compacted in some way that is making the device lighter and more comfortable to use as a regular glasses also the main frame can be removed in order to adjust other type of frames of your choice. Figure 2.18 shows the internal components of the mentioned device.

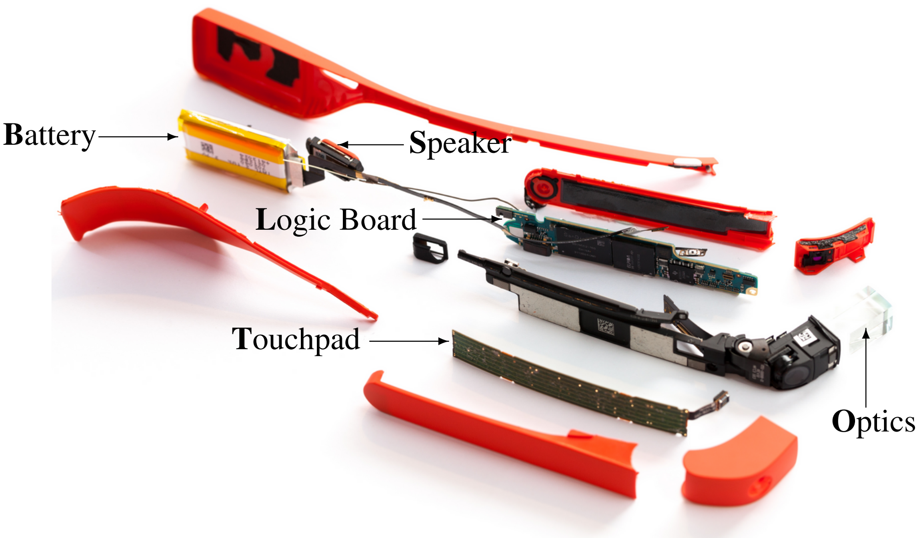
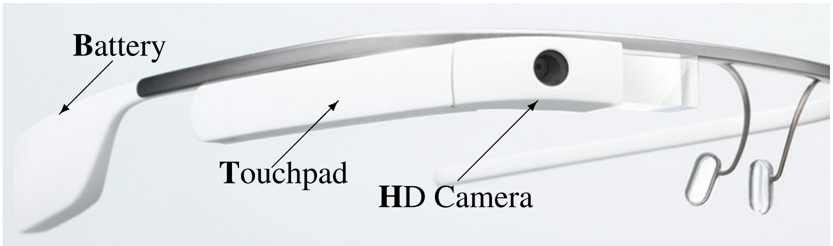
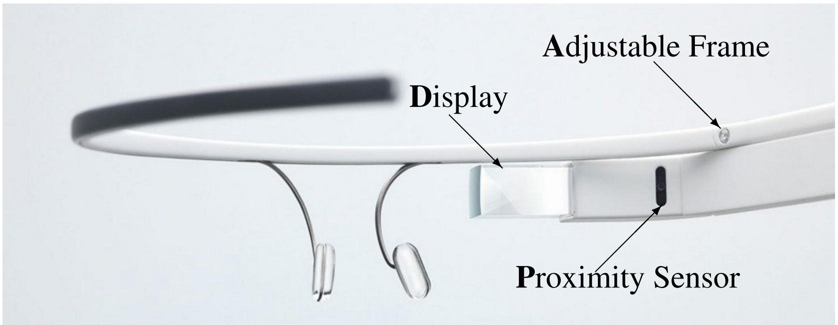


Figure 2.18: Project Glass Internal Components Description [TS13].

Finally, Project Glass was integrated by different elements included software provided by Google in order to deliver better experience to the final users. Figure 2.19(a) shows frontal view of the new design of the Google glasses making a huge difference in size in comparison with the prototype mentioned. On the other hand, Figure 2.19(b) shows another view of the Goggle Glasses in order to see the rear elements.



(a) Project Glass First Stage Prototype



(b) Project Glass Prototype press release Google I/O 2012

Figure 2.19: Pictures taken Project Glass web [Gog12]. (a) one of the first prototypes from Google (b) Prototype press release Google I/O 2012.

2.1.4 Virtual Reality Programming

In order to create VR systems, is needed the integration of programming components, or authoring, a virtual environment. The definition of VR programming changes depending on the point of view of the application. For VR system is an application programming interface Application programming Interface (API)-based method of creating a virtual world [BC03]. This section describes briefly the compilation of technologies working with APIs for VR systems.

To generate environments for VR systems is used flexible programming environments, it is also possible use low-level tools graphic languages (such as OpenGL). OpenGL is a trial-error iterative process that currently with the effort of the VR industry could be replaced by another software tool, nowadays called VR toolkits.

A toolkit is an extensible library with signed for creating VR environments, including the simulation of the parts and objects classifying attributes and classes, that simplified the programming tasks.These libraries exist since the first VR system developed in the early 1990 [BC03] until then these toolkits have been developed with different purposes.

WorldToolKit is one the oldest toolkit for VR programming introduced in 1990, supporting most commercial I/O devices such as trackers and Sensing Gloves. Then, technologies as WTK Scene Graph appears to give different options to create virtual environments in VR systems allowing multiple displays and windows to coexist.

On the other hand, is possible find in different scenarios the modalities of these VR toolkits. For example, the case of Java 3D which is similar to WTK, Java 3D uses OpenGL and Direct3D as a low-level graphic library functions and after years of research the previous toolkits were used in many applications and devices to represent the experience of VR systems.

Currently, toolkits such as Vizard VR integrates a solution to creating environments in VR. The toolkit is created by WorldViz and is a multipurpose VR development platform for building, rendering, and deploying 3D visualizations. In addition, this toolkit can be integrated into environments integrating different elements in one single platform. So far, the development of these platforms is huge as a customer product or as an integrator with new trends of technology.

Meanwhile, many technologies have been started to integrate different solutions to generate 3D objects and interactions between them, with the virtual environment including animations and effects pre-defined in the platform. At the same time, these tools started moving towards to different domains. For example, the 3D VirtualComponent factory Visualization tool that simulates manufacturing production lines in run-time. The basic principle of this visualization tool is the generation of 3D environments for industrial environments including humans, robots, conveyors, storages, materials and products.

The purpose of these visualization tools is to prevent accidents, save time and money at the moment of the real construction of the production line. The main costumers of these products are companies that build huge production lines around the world, these tools can help to predict future problems and integration with new manufacturing layouts. Figure 2.20 shows a picture taken from the main brochure of the company and is an example of the elements created by the 3D factory visualization tool.

Figure 2.20: Visual Component 3D Software Visualization Tool [Com99].

In the end, the creation of a comparison between the tools described before is required, taking under consideration new options available in order to compare and select the correct toolkit for the creation of the VR system. The toolkits can vary depending on the internal and external capabilities. For example, the programming languages used and the operating system where can be installed.

To select the right 3D engine for the creation of a VR system is needed a detail list of the requirements of the system also a description of the scenario where will be implemented including hardware and software available. For example, UNITY3D is a game engine software developed with different purposes including the generation of 3D environments for different platforms using different programming languages. UNITY3D can be installed in different Operating Systems (OS) and is also considered as an IDE.

In conclusion, UNITY3D has many capabilities, is multi-platform and multi-language engine that can create rich content applications for different platforms such as Linux-based Operation Systems for Mobile Devices (Android), iOS, PC and Mac. in addition, can be compatible with other formats from different vendors like CATIA or Open-Source 3D Computer Graphics Software (Blender). Figure 2.21 shows an example of a UNITY3D design for a futuristic workshop including the physics of the scenario. The complete example can be downloaded from the main page of the Asset Store in UNITY3D [Tec05].

Figure 2.21: Futuristic workshop/Robot Laboratory created in UNITY3D [Tec05].

2.1.5 Human Factor in Virtual reality

Until this point the architecture of the VR systems is described also each module of the architectures was introducing their own trends of technologies and explaining the basic functionalities and how the market is accepting it. Finally, the user is one of the main elements of the VR architecture described at the beginning of this Chapter. For this reason, this section describes how the user response based on the outputs of the VR systems.

The VR system starts measuring the user's performance when the simulation starts. Furthermore, it is necessary to understand why some user's responses lead to simulation sickness identifying which are the causes and factor that originate this effects. Finally, is studied and reviewed what kind of tasks could be done for minimizing its effects including the negative and positive effects that VR systems can have with regard to the society at large. The most current example is the expected release of Project Glass that based on the capabilities of this device in some places this device is forbidden to use.

The human factors for VR systems consist in a series of experiments, performed under very rigorous conditions, aimed at determining user's response to the technology usability, user safety, and the related societal impact of VR [BKLP04]. The studies related with human factors in VR can be added in a documented experimental protocol, these structured can consist of several session and periods. Moreover, the participants of these tests can be divided by gender, age, studies and factors that can affect the test in order to acquire better results.

Finally, as is predicted the acceleration of development in new trends of technologies change our everyday life and the sociological impact will be felt in three areas: the professional life, private life, and public life. In contrast, the users wonder what are the effects that the VR simulation can have in short or long periods of time, in short the effects of VR simulations on users can be divided in indirect effects and direct effects. The indirect effects are the neurological, phycological, sociological, or cybersickness (most known as the exposure to virtual environment). For example, a virtual game environment can cause an addiction or lack of real human relationships. The direct effect is mainly caused in the user's visual system also the user's auditory, skin, and musculoskeletal systems.

2.2 Augmented Reality

The domain of AR is intertwined with different technologies and emerging concepts. AR is multifaceted topic in different terms as a technology or group of technologies. The AR interfaces have been defined as a superimpose virtual information on the real world combining physical objects in the same interaction space, used in real time and are spatial [BKLP04]. Moreover, due to the development acceleration in VR systems, now AR is considered as an instance of VR involving the computer graphics overlaying the vision of the user, this immersion of VR is now applied for different applications.

This section described the result of the large development of AR systems. Thus, fallowed by the description of MAR considered as a new special instance of AR. Various elements are behind MAR and application areas in which MAR is applied and described. Finally, an overview of the user experience and human factors is conducted in specific areas of industrial systems especially in the domain of manufacturing systems.

2.2.1 Introduction to Augmented Reality

The emergence of AR occurs with the growing implementation and development of the concept of VR. Since, VR immerses the user in a computer-generated world whereas AR combines the real world with computer graphics, mixing the physical reality with the virtual content [Mad11]. The AR general principal lead in some layers such as the capture, the register and the Augment [BKLP04]. In other words, is the acquisition of signals from the real world, the transformation of those signals to display virtual content in front of the user without losing the context of the real environment, melding or replacing the real with the virtual objects. The basic description of AR and three elements mentioned are always present in AR systems.

Currently, AR applications reached great expectations about the advantages of this technology. In particular, taking into consideration the principal advantages of the AR applications which is the mobility and the fact that almost everybody has a mobile device. Moreover, the fast growing up of development in mobile devices has been changing the methods of programming applications, now the creation of a mobile application is easier with more friendly APIs.

The fast development of new Hardware and software allows the companies and developers the creation of projects integrating new trends of technologies applying the concepts of VR, AR or MAR. As a result, Figure 2.22 describes these trends of technologies in a prediction of couple of year. The chart shows a study of more than 1,900 technologies predicting and expectation in domains including Hybrid Cloud Computing, HTML5, wireless Power and NFC payments are moving towards.

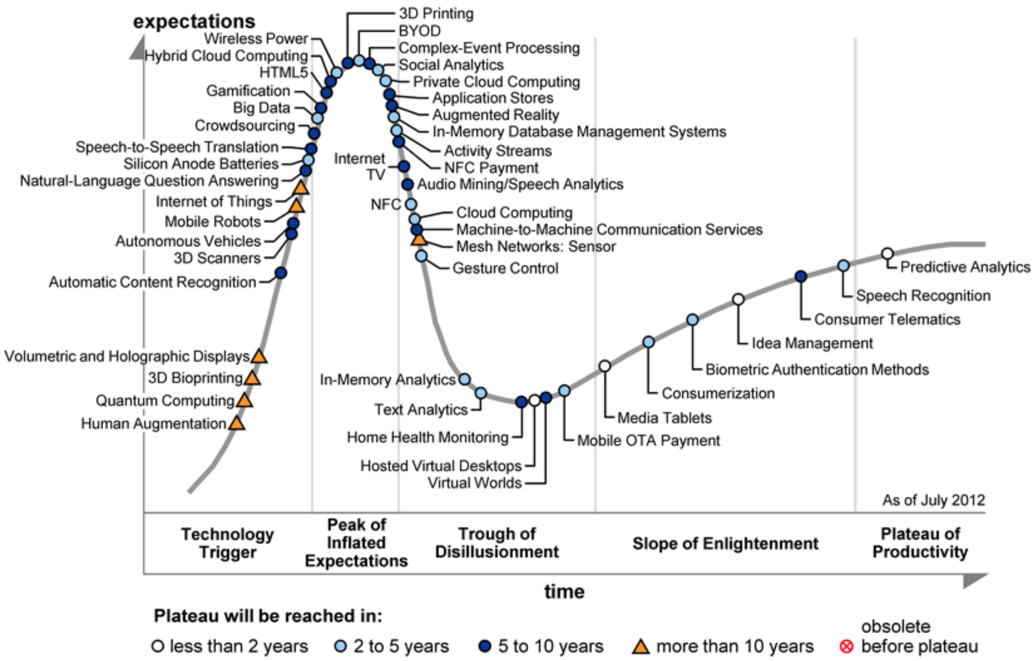


Figure 2.22: Study of Overhyped: Hybrid Cloud Computing, HTML5, Wireless Power, NFC, AR, Mobile devices and Wearable devices [Gar12].

For this reason, the competition had started with the development of smaller and faster devices with an integration of awesome software including voice recognition, eyes tracking and AR.

In 1999, the first demonstration with AR arrives at the market using components as the camera position, tracking codes and the ability to use any square marker pattern. The library ARToolKit was originally developed by Dr. Hirokazu Kato starting given support only for windows PCs [Kat99] and its ongoing development is being supported by Universities and companies.

Now, after couple of years of development, different companies start developing in this platform integrating more elements and creating private software for different applications. For instance, companies like Apple, Google and Microsoft compete in this and others fields of VR to create better applications based on AR, Internet TV and cloud computing. In particular, AR, applications are implemented in many fields and strongly influenced in mobile devices. The aim of the AR applications is to show objects that help to the user to have a better perception of the reality helping the user with extra information that can be useful to have in that precise moment. The AR applications use images from the real world and 3D objects to interact with the user.

2.2.2 Mobile Augmented reality

As a result of the acceleration of development and market sales related mobile devices, AR was also incorporated into the mobile device domain. In the beginning, applications were developed for desktop computers using USB web cameras to create the experience of AR based on the same principles of AR using cheap hardware and open source libraries this technology starts moving towards. Figure 2.23 shows the application based on AR using a desktop web camera and ATOMIC Augmented reality and Mapping [Lib09].

ATOMIC was created based on libraries from ARToolKit the application has the objective of create basic examples in AR without knowledge of programming or expertise in VR systems. In comparison with ARToolKit, this software is also open source and is a good start to test basic examples of AR. In addition, the latest update of ATOMIC is included the integration of web development, in other words, the deployment of the examples used in this software for WB.

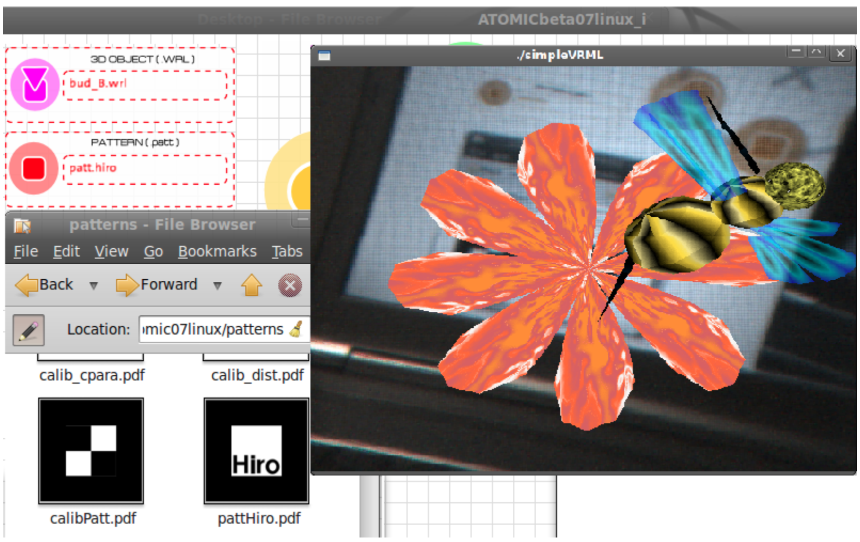


Figure 2.23: ATOMIC Authoring Tool for Augmented Reality (AR) and Mapping.

Therefore, was a matter of time that applications based on libraries of ARToolKit start the emigration to other platforms. Thus, the first AR application appears for mobile devices, as an illustration, the first demonstrator using MAR was an indoor guidance system, using low-cost mobile devices, the application is able to read several targets displaying 3D object element in the screen of the mobile phone [Wag07]. Meanwhile, the new era of the smartphones starts equipping new devices with sensors, GPS and better quality cameras. Moreover, regarding to integration of sensors in mobile devices such as compass, accelerometer and gyroscope, these combined in a single chip the user can experiment different sensations now called Sensor Fusion. In brief, Sensor Fusion incorporates the elements of the sensors increasing the accuracy and response, making the application more precise when a user action is performed. For example, Micro-Electro-Mechanical Systems (MEMS) technologies created by InvenSence that are supported and integrated into products of companies such as Nintendo and Samsung. In addition, InvenSense is one of the biggest distributors of MEMS technologies, working in collaboration with several companies to deliver a better experience using this micro-sensors with 6-axis and 9-axis also developing software applications integrating Sensor Fusion, in order to deliver more faster application. The following Figure 2.24 shows the Motion Processing Library into MEMS using in applications such as smart TV, VR, AR and MAR. The library simplifies the integration and linking of the application processor, making the access to the motion data in the form of quaternios, rotations, matrices including the measures as the linear acceleration and gravity. The library, basic examples and manuals are available for developers and companies interested in build application based on MEMS.

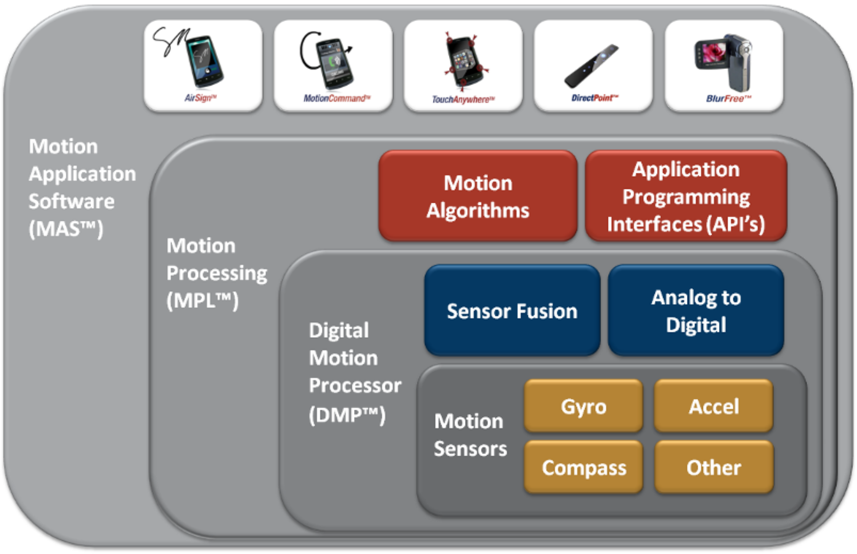


Figure 2.24: InvenSense: Motion Processor Library [Inc11].

Finally, it is a fact that the user is involved in an environment of technologies. However, sometimes the user does not know the real use of some devices and they only cares about the results of the actions performed by the device, which is basically the time that the user will expend using the device. For this reason, the user expects better devices with new services including the performance of new applications mixing software and hardware to deliver a quality product. For example, the integration with the new internet services and the developments of cloud services in order to process and share information with other computers without affecting the performance of the device. For this purpose, the WB development starts moving forward to the era of cloud computing including sensors such as MEMS and GPS delivering more options to create new applications integrating other technologies. Currently, companies such as Layar, Wikitude, and Junao are making easier the development of AR applications, in particular for mobile devices. The integration of WB in applications for AR changes in some way the traditional acquisition of images from the real world.

In order to achieve better results in AR, is important to remember the main elements of AR. For this reason, the following modules are described with more details. The capture is the acquisition of images from the real world, for the first demonstrator with AR they start using simple patterns in order to make easier the registration of the image. The AR markers are considered as one of the main elements in AR applications. However, in order to have a better performance in the AR applications it is needed a high quality marker (unique and robust), the properties of the AR markers lead in the Table 2.1 describing some of the main properties of the markers for AR.

Table 2.1: Augmented Reality (AR) Markers properties.

Properties	**Description**
Easy to IDENTIFY	Can be recognized in scene of real object. Multiple AR markers can be used simultaneously.
Easy to TRACK	Works with noisy camera image. Not effected by external factor as the light.
Easy to PROCESS	Determine camera distance. Determine camera angle. Determine camera rotation.

On another hand, these patterns have been changed in different forms from simple black and white images to different types of RGB images. The tracking algorithms are also linked to the patterns and have been improved creating more complicated the tracking of elements, increasing the detection of edges and special features inside the image. In addition, there is a long way to improve the mentioned elements in AR. Currently, the algorithms are more stables in such way the present libraries in the market can detect features from images taken from the real environment such as books covers, CD covers and buildings.

The evolution related to the tracking algorithms is called Natural-Feature tracking. The Natural tracking expands the concept of a AR markers. In contrast to the traditional markers the Natural tracking works in different ways and can be applied for any image as long the image is complex enough. In comparison with the Table 2.1 the Natural Tracking is also leaded by these three rules. However, these solutions are classified in two branches. The marked-based AR are applications based on patterns defined by the system, Figure 2.23 described in previous section is a graphic example of a marked-based AR application. Then, the Figure 2.25(a) shows the classic marker-based image from the first library developed by Dr. Hirokazu Kato. On the other hand, Figure 2.25(b) shows and example of a marker-less AR application using the box of a product to display the content of the box before the purchase.

(a) Marker-based AR application. (b) Marker-less AR application from LEGO stores.

Figure 2.25: Pictures taken from web sources of traditional applications using Marker-based and Marker-less tracking objects. (a) Computer Tracking Library of Augmented Reality (ARToolKit) Marked-based application. (b) LEGO stores, customer using AR before the purchase using the cover of the box.

In addition, due to the integration of AR in mobile device also composition of WB applications with AR, is making easier the acquisition of applications with AR. As a result, AR technologies are changing the mobile device market sharing with many other trends of hardware and software technologies creating more attractive applications. Nowadays, the web AR applications are emerging delivering different views of what the user can do with MAR applications. Figure 2.26(a) shows the application after scan the environment around the user. The application used web technologies including the elements of AR such as the capture, registration and augmentation.

The images are taken from the real world and analyzed by Metaio engine, then displaying relevant information that the user probably could use. In comparison, Figure 2.26(b) shows an example of a GPS driver assistant based on AR, using the camera and sensors of the mobile device, this application can deliver different experiences using the integrated smartphone with GPS and Sensor Fusion technologies, now the user can see within real images which street needs to take to turn right or left.

(a) Metaio application using Natural tracking to detect near buildings around the user [Met03].

(b) Wikitude Augmented Reality (AR) application Driver assistant [Wik08].

Figure 2.26: (a) Metaio engine. Augmented Reality (AR) application for Tracking environment. (b) Wikitude Augmented Reality (AR) application Driver Assistant.

2.2.3 Building Augmented Reality

The support for developing applications in AR is huge comparing to others APIs in the market and there are more options that can be selected based on the requirement system including the users and environment. The AR applications are supported in different OS such as Android, iOS and nowadays WB. In addition, the references related AR are huge enough to make decision and comparison between the APIs implemented, taking examples from different perspectives (see Table 2.2). For instance, complete manuals in books as [Mad11], that make a brief comparison between the history of AR and the technology nowadays implemented using the integration of applications for Web Browser (WB) with AR. The author makes an easy guide of development starting a basic application with AR using different APIs for example Layar, Wikitude or Junaio, introducing a detail description of each API with theirs components, how they work and how it is possible generate new applications no matter which OS is used in the mobile devices.

However, in comparison with Tony Mullen [Mul11] the book describes the steps to how create your AR application from the basic concept of AR. In simple terms, using source libraries from AR, the user can generate a marker-based AR application from the source code. In fact, it is needed the advance knowledge of Image processing and programming. In contrast, in [Mul11] the entire concept of AR is explained, guiding the creation of 3D objects in open source software tools for creating a full open source AR application project. As a result, the path to build an AR application could lead to many factors, but it is impossible denied the existent of different options to create awesome AR applications. This section at the same time includes some of the options available to build rich content application based on the concept of AR.

1. **Computer Tracking Library of Augmented Reality (ARToolKit):** Is a software library for building Augmented Reality (AR) applications. These are applications that involve the overlay of virtual imagery on the real world [Kat99]. ARToolKit uses computer vision algorithms to solve the problem of tracking a viewpoint, calculating the real camera position and orientation relative to physical markers in real time. Ideal if the project application starts from the source code. Some of the features of ARToolKit are:

   - Single camera position/orientation tracking.
   - Tracking code that uses simple black squares.
   - The enable to use any square marker pattern.
   - Easy camera calibration code.
   - Fast enough for real time AR applications.

- Multiplatform Unix Operating Systems (Linux), MacOS and Windows OS distributions.
- Distributed with complete source code.

2. **Qualcomm Vuforia Augmented Reality (VAR):** Is a software platform that uses top-notch, consistent, and technically resourceful computer vision-based image recognition and offers the widest set of features and capabilities, giving developers the freedom to extend their visions without technical limitations. With support for iOS, Android, and UNITY3D. The VAR platform allows you to write a single native applications that can reach most users across the widest range of platforms including smart devices and wearable devices [Qua12]. Figure 2.27 shows an image example of the integration between VAR with UNITY3D.

Figure 2.27: Using Vuforia Augmented Reality (VAR) library to integrate UNITY3D. Image taken from Vuforia developer web page [Qua12].

3. **ALVAR:** Is a software library for creating virtual and augmented reality applications. ALVAR has been developed by the VTT technical Research Center in Finland. The library includes a high-level API and several tools for creating Augmented Reality applications with just a few lines of code [oF12]. At the same time, this library can render 3D and 2D Images from the images captured by the device. Figure 2.28 shows an example using ALVAR library. The following list contains the main features of ALVAR based on the type of AR application.

- **Marker Based Tracking**
  - Accurate marker pose estimation.
  - Two types of square matrix markers.
  - New marker types easy to add.
  - Recovering from occlusions.
  - Multiple markers for pose detection.

- **Marker Less Tracking**
  - Feature-based (tracking features from environment).
  - Template-based (matching against predefined images or objects).

Figure 2.28: Using ALVAR Augmented Reality library. Image taken from ALVAR brochure [oF12].

4. **AR Web Browsers:** The development of AR application starts moving to the direction of Web Development. For instance, Layar, Junaio and Wikitude. However, these technologies have their own advantages and drawbacks and the following list describes the elements needed to be taken into consideration:

   Benefits of using AR browser platform:

   - Build content with little or no programming knowledge required (data bases providing XML to reference the content).
   - The content appears to an audience already interested in AR and actively seeking the content.
   - You do not have to worry about the camera input, screen layout, or GPS APIs.
   - The same content work for Android, iPhone, and Symbian/Windows. Phone/bada, OS/MeeGo when the client are released.
   - Common for AR browsers to be preinstalled on mobile devices. [Mul11].

   Drawbacks include:

   - User must launch another application before finding the content.
   - The content can lay undiscovered in the platform provider's client.
   - You don't have full control over the user interface. [Mul11]

AR Web Browsers

Each AR browser offers unique functionality. As a matter of fact, the following list integrates the three most used AR browsers into the market.

- Wikitude: The first Augmented Reality browser appearing for Android devices in 2008 [Jun09]. The Wikitude World Browser displays information about the user's surroundings in a mobile camera view. The position of the objects on the screen of the mobile device is calculated using the user's position (GPS or Wi-fi), the direction in which the user is facing (by using compass) and accelerometer [Wik08].

- Layar: Specializes in mobile augmented reality - the most popular medium through which the average person interacts with AR content. The mobile AR world consists largely of two different types of experiences: geolocation- and vision-based Augmented Reality (AR) [vdKBLF09].

- Junaio: Is the easiest entry point to developing and publishing augmented reality and location-based experience. The Junaio framework works for Android and iOS devices, Junaio have several million users and a thriving international and professional developer community [Jun09].

Finally, note that current AR applications are more commonly used and the market and have a great repertory of applications including open libraries and code that could be used for testing purposes or integration into other application. In conclusion, this Section was described the relevant content of the different platforms and libraries to create AR application making a comparison with the different technologies and applications in the market. Thus, Table 2.2 describes the technologies mentioned in this section, compiling the information making easy the selection of a technology, platform or library for future developments.

Table 2.2: Technologies Building Augmented Reality.

Software Library	Application Type	AR Type
ARToolKit	PC Mobile	Marker Based Tracking
ALVAR	PC Mobile Web Browser	Marker Based & Marker Less
Vuforia	PC Mobile Cloud Tracking	Multiplatform library Graphical interface Marker based &Marker less.
String	Mobile	Only for iOS Marker Based
Web AR	Web Browser Sensors GPS	Used only for web browsers position and localization Marker Based & Marker less

2.2.4 Approaches of Mobile Augmented Reality

This section integrates the mentioned concepts along Chapter 2 related with Augmented Reality (AR), theirs instances and applications. Currently, the AR is present in many domains and this section contains examples of integration with AR in the different domains, describing the areas where MAR is implemented taking some references and the conclusion with the integration in manufacturing systems.

Medical

In the current Medical domain, it is not surprising that this domain is viewed as one of the more important for Augmented Reality (AR) systems. Most of the medical applications deal with image-guided surgery. Pre-operative imaging studies, such as CT or MRI scans, of the patient provide the surgeon with the necessary view of the internal anatomy [SVH+02]. From these images, the surgery is planned and is possible to have a visualization of the path through the anatomy to the affected area where, for example, a tumour must be removed is done by first creating a 3D model from the multiple views and slices in the preoperative study [SVH+02].

Manufacturing, Maintenance and Repair

When the maintenance technician approaches a new or unfamiliar piece of equipment instead of opening several repair manuals they could put on an AR display. In this display the image of the equipment would be augmented with annotations and pertinent information to the repair [SOG03].

Boeing researchers are developing an augmented reality display to replace the large work frames used for making wiring harnesses for their aircraft. Using this experimental system, the technicians are guided by the augmented display that shows the routing of the cables on a generic frame used for all harnesses. The augmented display allows a single fixture to be used for making the multiple harnesses [SOG03].

Robotics and Telerobotics

In the domain of robotics and telerobotics an augmented display can assist the user of the system. A telerobotic operator uses a visual image of the remote workspace to guide the robot. Annotation of the view would still be useful just as it is when the scene is in front of the operator. If the operator is attempting a motion it could be practiced on a virtual robot that is visualized as an augmentation to the real scene. The operator can decide to proceed with the motion after seeing the results [SOG03].

In conclusion, the implementation of AR can vary depending on the used technology and the scenario where is applied. The mentioned examples clarify just the concept of AR applications and theirs implementation in different domains. The following Chapters describe with more details the AR integration in a specific domain in particular cases.

3. APPROACH AND METHODOLOGY

This Chapter describes the methodology and techniques to analyze, design and implement an application in AR. Due to the continuous quality improvements, in this section, the life cycle of the application change in order to develop a high level software solution. The selected methodology takes into consideration factors such as the environment and the user. However, to achieve the objectives presented in this document is needed the organization and planning of each task starting from the acquisition of the requirements until the test of the first stable version of the AR application. Augmented VIsualization for Transparent Factory (AVITA) is created based on the analysis, design, implementation and testing phases described in the present Chapter 3. In addition, the use case applied in industrial systems is mentioned, in order to acquire the necessary information to create a relationship with the first version until the last version of AVITA explained in Co-summit 2012 as a part of ASTUTE project. In general, the methodology used to control the versions of this software is the Iterative/Spiral development methodology (see Figure 3.1).

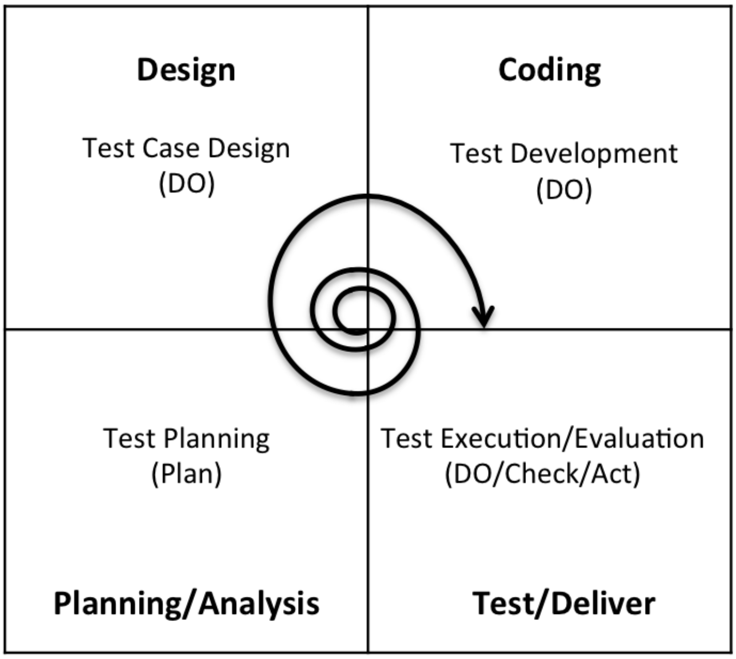


Figure 3.1: Iterative/Spiral Software Development Methodology [Lew09].

In contrast to other methodology techniques, the Spiral methodology is a reaction to the traditional waterfall methodology of systems development. However, the waterfall methodology is used at the beginning of development ofAVITA, as a sequential solution development approach to integrate the first version. Moreover, due the results of the first version of the application, the workflow continues changing to the Spiral methodology reaching more objectives described in the initial state of AVITA.

In the initial state of ASTUTE project were defined some main modules in order to design the system. In particular, the modules of 3D and AR are integrated into the main solution of ASTUTE in the side of the visualization part to the creation of HMIs. On another hand, it is shown in the Figure 3.1 the application starts from the Planning and Analysis of the system requirements, then designing the system, implementing translating the design models into code and finally the test and deliver of the application.

One of the advantages applying the Spiral process is that the clients receive at least some functionality quickly. Another benefit is that the product can be shaped by iterative feedback, for example, users do not have to define every feature correctly and in full detail at the beginning of the development cycle, but can react to each iteration [Lew09].The following list describes with more detail the phases of development of AVITA, describing the Spiral development. In conclusion, the Spiral sequential development expedites product delivery. A small but functional initial system is built and quickly delivered, and then enhanced in a series of iterations.

1. **Planning&Analysis:** The spiral process starts in this phase, including the use case with the objectives of delivery and contains a set of functionalities of the software. The result after the interaction of the Spiral process is shown in Figure 3.4 describing the functional and non-functional requirements. Thus, Figure 3.5 shows the final results of the iteration with the Planning & Analysis process.

2. **Design:** The design process in the sequential Spiral methodology is created based on the components of the system and delivered functionalities by the Planning & Analysis process. As a result, more diagrams are created in System Modeling Language (SysML) in order to model the system. The Figure 3.7 shows the Use Case diagram based from the general abstraction of the system. In the same way, Figure 3.8 shows another use case diagram, created based on the final interaction with the sequential development process.

3. **Coding:** In this phase the functionalities of the application are created based on the System Modeling Language (SysML) diagrams mentioned above. This particular phase is described with more details in the Chapter 4.

4. **Test/Deliver:** This phase is described with more details is Chapter 5. containing

the iteration that the software had with the user along the spiral process. Also the different versions of AVITA are mentioned in this phase including the current state of development.

Afterwards, in order to start with the Spiral process a scenario description is needed. In Section 3.1, the scenario is described and a text description is taken as a reference to retrieve functional and non-functional requirements.

3.1 Scenario Description

The industrial domain has been changing the methods of manufacturing and in the same way, new trends of technology are integrating hardware and software in the industrial domain with the objective to help the users to make tasks easier in a normal situation of event occurrence. Moreover, this scenario describes an industrial manufacturing production line continuously producing parts, the production line contains several elements such as robots, sensors, motors, conveyors and pallets. In addition, there are different users involved in the system in charge of different daily tasks.

The operator user has a role important in the layout of the manufacturing controlling and checking the production line all the time, taking into account elements such as the performance and quality inspection. The maintenance user is called when an event occurs related to malfunction in the production line or an error that the operator cannot fix and needs special attention. Finally, the management user can handle the material and human resources including the complete access to the system and visualization of the general information about the production. For instance, the status of the production, KPIs and general information.

On the other hand, Figure 3.2 shows the physical implementation of production line and some of the elements that integrate the manufacturing layout. The real location of the manufacturing system is in Tampere University of Technology, Factory Automation & Technology Laboratory. The same scenario is applied to some projects in the department and is the current physical layout for different demonstrations, included the manufacturing use case for ASTUTE project.

The scenario is composed by physical and logical modules. The physical scenario is composed by ten cells and each cell contains a robot with a PLC, several motors, sensors and actuators. The pallets are transported by several conveyors including a bypass conveyors if a station is busy or in maintenance mode, the entire layout have temperature sensors, proximity sensors, emergency switches and devices with Radio-Frequency Identification (RFID)s to detect the position of the pallets.

The logical model is represented by logical inputs and outputs generated by the PLCs, signals that are processed by the PLC and sent to the system. In the end, the information is gathered, stored and processed by the system in order to have the current status of the system. As a result, different assumptions are made in order to define the requirements of the systems and how AVITA will fulfill those tasks.

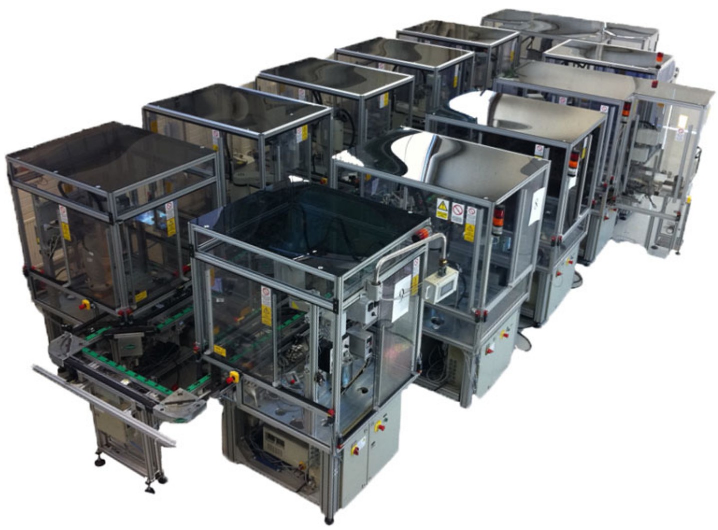


Figure 3.2: Scenario Description FASTory.

3.2 AVITA: Sequential Development Architecture

This section describes with more details the sequential proposed process in the development of the application after and before the first stable version of AVITA. In the beginning, a software engineering tool was selected in order to create a visual representation of the structure of AVITA. Currently, there are different options to create a visual representation of the project such as Visual Paradigm for UML [Int12] or IBM Rational Software [IBM81]. In contrast to mentioned tools IBM Rational Rhapsody provides a collaborative design, development and test environments for system engineers and software engineers, with rapid prototyping and execution to address errors earlier when they are least costly to fix.

In conclusion, IBM Rational Rhapsody is used in the sequential process of the development of AVITA, initially for modeling the Analysis and Design process. The IBM Rational Rhapsody is a modeling environment based on Unified Modeling Language (UML), is a visual development environment for system engineers and software developers. The purpose of the modeling environment in this application was the creation of diagrams in SysML as a derivation of UML.

The diagrams have been changing across the sequential process (Spiral development Figure 3.1). However, the main objective of this section is the detailed description of some mentioned process such as planning and analysis Section 3.2.1 and application design Section 3.2.2. As a result, the different versions of the logical architecture of the system were created including the different tests of AR in an initial and final stage. Figure 3.3 shows the initial view of the system integration with the main objectives of this document. The creation of an application applying the concept of AR in industrial systems.

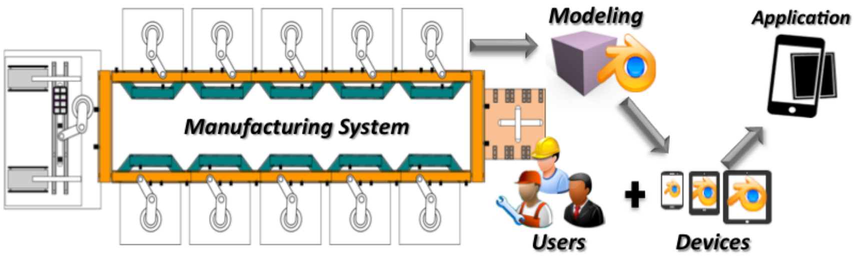


Figure 3.3: Layout System Integration.

3.2.1 Planning and Analysis

This subsection describes the method that was used to retrieve information from the scenario description including the main functionalities of the application based on functional and non-functional requirements. The method starts the Planning and Analysis process. The process begins in the analysis of the principles of Augmented Reality (AR). The concept of AR and the description of the scenario are translated into a modeling language. In this example, SysML models, in order to generate the initial requirement diagram. On the other hand, as was mentioned in the sequential development approach used in this MAR application, the first delivery version of the requirement diagram was improved in a short period of time based on new incoming requirements. Then, different versions of diagrams were created and some of them are described in this document in order to briefly describe the Planning and Analysis process.

Requirement Diagrams

The initial requirement diagram created in SysML contains the most general elements from AR and their integration with MAR. For instance, the Capture, Registration and Augmentation. The general elements are taken from the analysis process containing some technologies with their functionalities. Then, the scenario was included in the process taking into account the general functionalities that the system should be achieve. Figure 3.4 shows the third version of the requirement diagram modeled in SysML with IBM Rational Rhapsody, Including the general requirements of the application also with the experience acquired along the implementation some new elements are integrated as the AR library and the 3D object requirements.

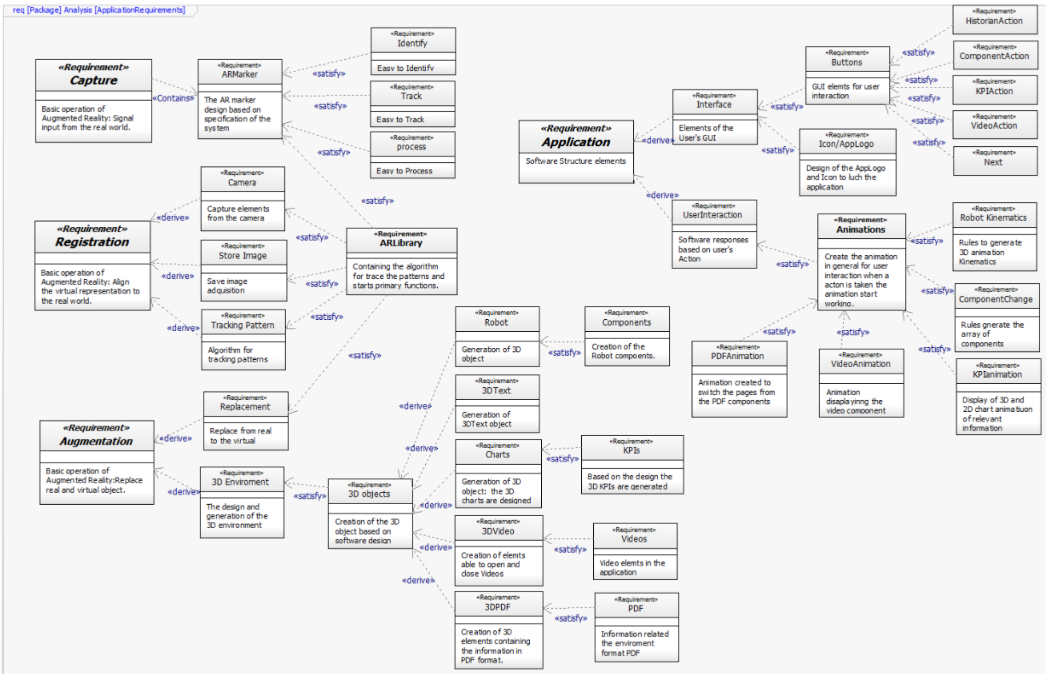


Figure 3.4: Requirements Diagram SysML Third Version.

After several interactions with the Spiral process, some of the diagrams have been changed and the requirement diagram was not an exception, due to the new improvements of the application some new requirements emerge and are added to the model itself. As a result, the final requirement diagram shown in Figure 3.5 includes the elements that integrate AVITA as an initial stable released including the hardware and software integration in different modalities of the user.

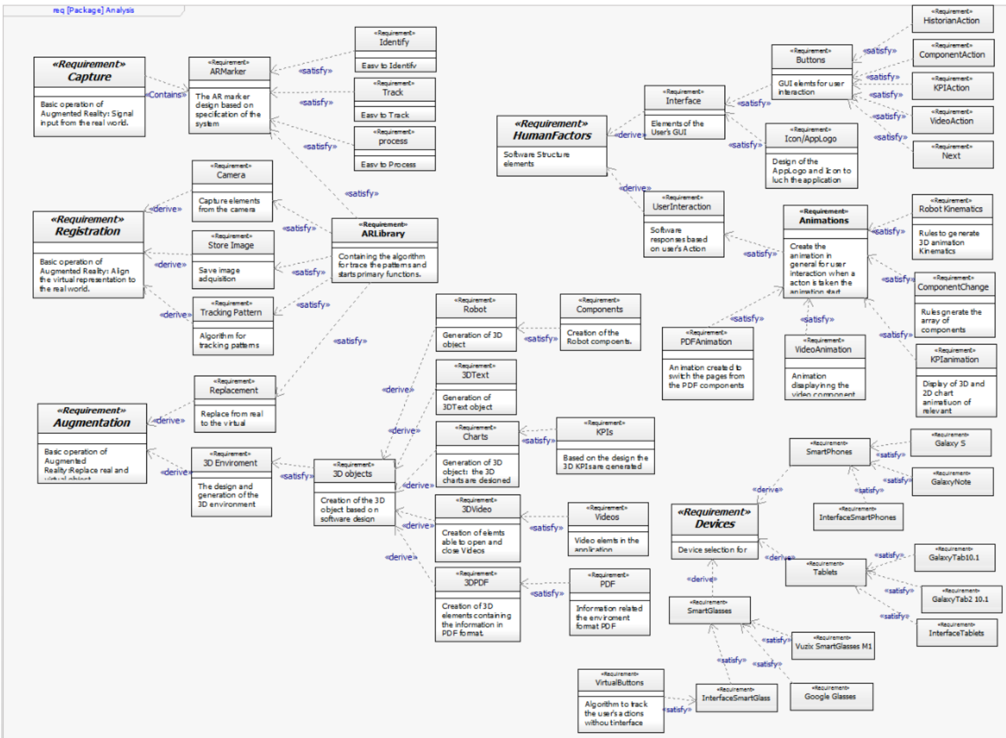


Figure 3.5: Requirements Diagram SysML Stable Release Version.

On the other hand, the planning process also creates the comparison based on the testing of several technologies (see Chapter 2). For example, Figure 3.6 shows the compilation of a basic example of an AR application using ATOMIC libraries. This application was installed with the objective of a close inspection of the primary elements of AR, compiling the results in the planning process for future development, in this case, the creation of AVITA.

Figure 3.6: ATOMIC Authoring Tool for AR and Mapping.

In conclusion, the requirement diagram illustrated in Figure 3.5 helps to the integration of the MAR application into other systems also if a developer team work on the same application it is possible see the integration of the internal and external functionalities in the application making easier the interaction with new modules.

Use Case Diagram

The use case diagrams were created based on the requirement diagrams shown in the Subsection 3.2.1. The use case represents the connection between the users and the main functionalities of the system. In this example, the interaction between the different users and AVITA. However, as it is mentioned in the introduction of Chapter 3 the Spiral process is a fast delivery and the interaction between layers change continuously. Figure 3.7 shows the first version of the use case modeled in IBM Rational Rhapsody.

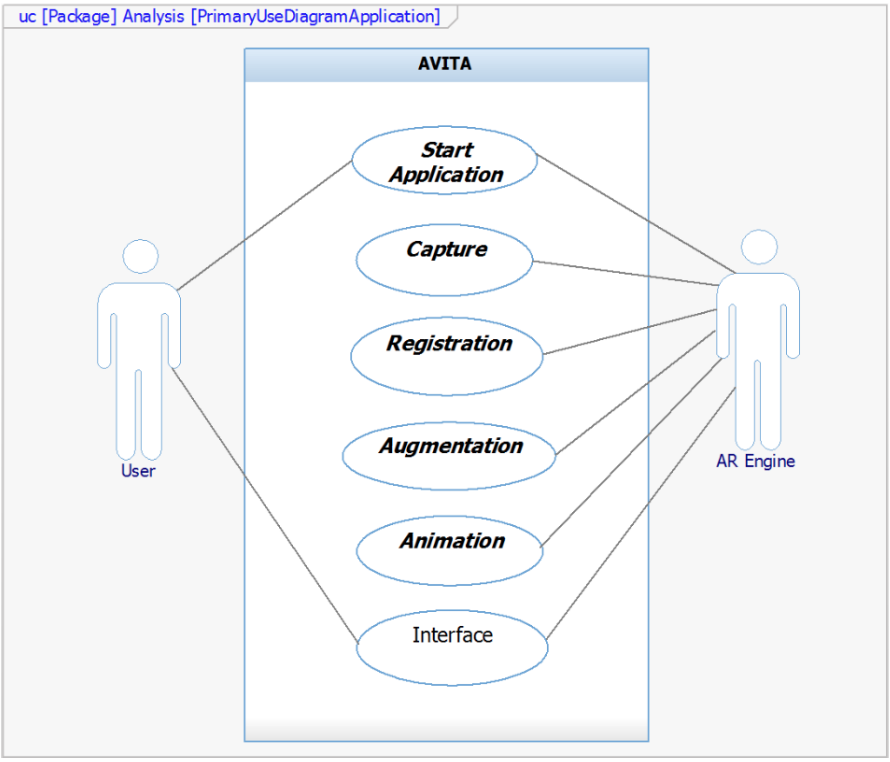


Figure 3.7: Use Case Diagram General Scenario.

In order to specify the actions in the use case diagram, the following list was created, describing the operations and interactions performed by the actors in the Use Case diagram.

- **Start Application:** The user selects the initial icon of the MAR application. Then, the AR engine launches the application based the features of the mobile OS.

- **Capture:** After, the application is lunched the AR engine calls the library for capturing images from the embedded camera in the device.
- **Registration:** The Registration starts when the Capture method sent the first set of images to the registration module. This method track the pattern previously designed in order to get the coordinates and position of the pattern.
- **Augmentation:** This operation is performed when the tracking algorithm sends the position and coordinates of the pattern, displaying 3D and 2D information overlaying the camera inputs.
- **Animation:** In comparison with other steps, this operation is not performed in the AR library. The animation is created based on the actions of the user. These operations are described with more details in the Section 3.2.2 & Section 4.
- **Interface:** This operation represents the interaction between the user and the application. The interface will be described with more details in Section 3.2.2 & Section 4.

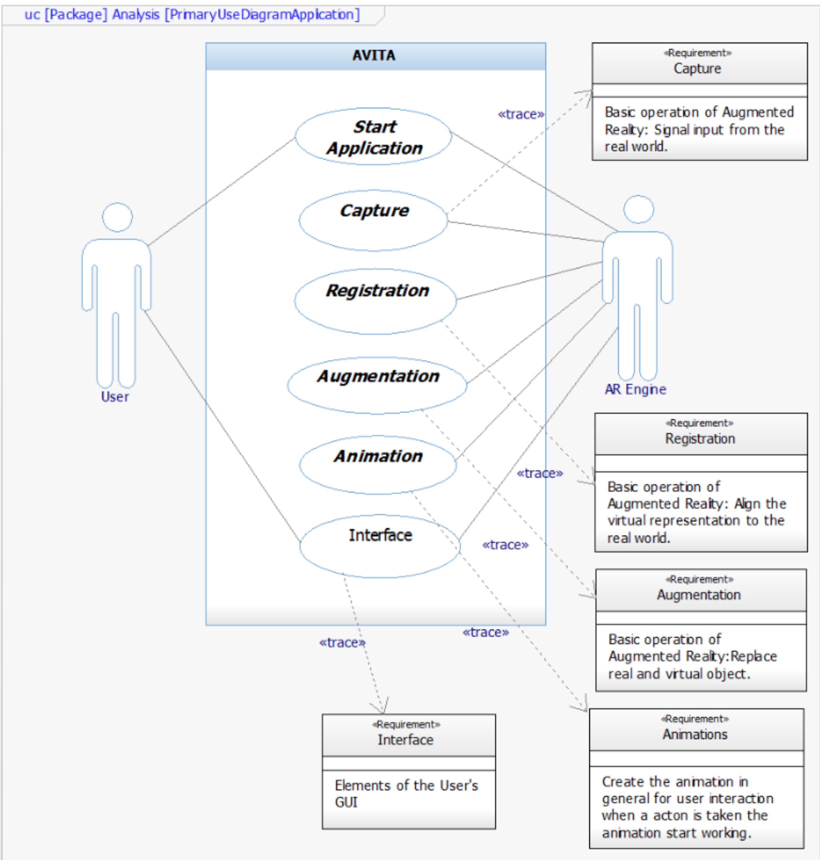


Figure 3.8: Use Case Diagram Relationship with Requirement Diagram.

In addition, as it was mentioned in Section 3.2 about following the life cycle of the application applying the Spiral process, Figure 3.8 creates the connectivity between the interaction user-action and action-requirements, using the «trace» in the modeling language (SysML) the operations are linked with the requirements. This means that the operation fulfills the functionality of the requirements connected in the Figure 3.5.

Visualization Draft

This subsection describes some of the elements that integrate the visualization part in the MAR application. At the beginning of the application, some elements were defined in order to decide what kind of elements are needed in the HMI. As a result, Figure 3.9 describes the first generation of elements that integrate the application and represent the elements from the Planning and Analysis process.

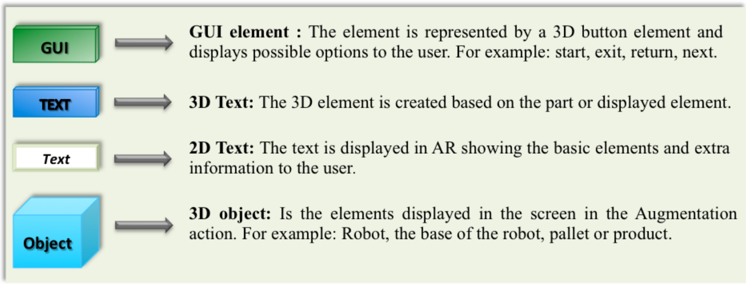


Figure 3.9: Planning and Analysis AR elements.

On the other hand, these elements were improved adding more operations and values to the system. For example, the reusing elements for other scenarios like the video element used for maintenance or operational purposes. Figure 3.10 describes new elements after the design process mentioned in Section 3.3. Then, the elements designed in this section are integrated into the interaction flow of the interfaces.

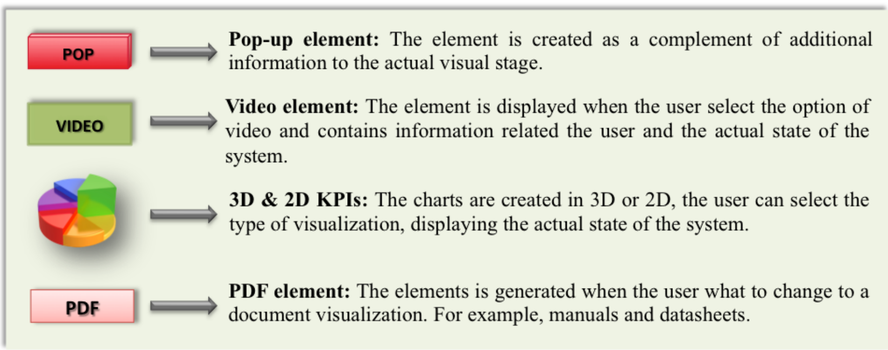


Figure 3.10: Planning and Analysis Integration after Design process (AR elements).

3.2.2 Design

This subsection describes the elements that integrate the design process from the logical point of view until the delivery of some graphical drafts to create the final graphic interaction with the user and the application. As a result of previous processes, the interaction flows continue after the first version of the Use Case (Figure 3.7). Based on the initial iteration with the design phase sequential diagrams were created in order to see in more detail the interaction user-application (see Appendix E). Figure 3.11 shows the general sequential diagram with the interaction between the user and the application including reference blocks that represent the connection with the other sequential diagrams describing with more details the data interaction between the user and AVITA (see Appendix E).

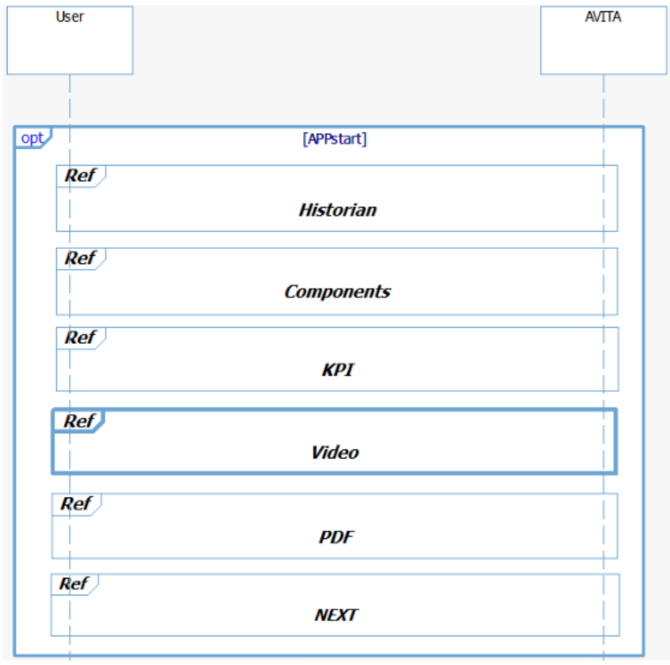


Figure 3.11: Sequential Diagram Interaction Description User-AVITA.

Furthermore, based on the logical described interactions from previous Section 3.2.1 and Section 3.2.2, several improvements are added to the architecture in Figures 3.4 and Figure 3.7 integrating more complex elements in the scenario based on the results of the testing process. Table 3.1 shows the new requirements added to the system, such as the user interaction with the Animation module. For example, the Maintenance user should be capable to seeing the current state of the robot including the components of the robot, also the manuals of each component can be displayed and videos explaining how to fix the component. In comparison, the operator should be able to see the simulation of the robot with historical information, videos about the behavior of the robots and safety manuals.

Table 3.1: Integration of complex user's scenarios

Scenario	Requirements
Management	Historian Performance KPIs
Maintenance	Robot current status Components PDF manuals Videos
Operator	Simulation Videos Operation Manuals Determine camera rotation.

On the other hand, based on the different user's scenarios, new hardware was included in the design process. Then, Table 3.2 was integrated with new device requirements in order to give more options to the user, no matter the type of used device in the manufacturing floor, for example, the integration of wearable devices such as the smart glasses. This action change the structure of the application because the design of AVITA is required for the different sizes of screens and the position of the interface will be adapted for the type of device. Thus, this action was taken in order to integrate AVITA for wearable devices such as Vuzix smart Glasses M100 and Goggle glasses. As a result, the Spiral process takes the new requirements of design and development.

Table 3.2: Integration of complex scenarios with new devices

Device	Requirements
Smarthphone	Screen OS version Hardware capabilities Interface
Tablets	Screen OS version Hardware capabilities Interface
Smart Glasses	Screen OS version Hardware capabilities Interface

Patterns Design

Further on, more elements of the AR application are required to be defined. In this part of design are integrated the AR markers starting with the creation process based on the new requirements shown in Table 3.2. As mentioned in previous sections the AR markers are one of the main elements that integrate AR. Then, to create an AR application is needed the design and test several patterns to find the appropriate pattern to be used for tracking.

The patterns are defined by the rules mentioned in Table 2.1 and Figure 3.12(a) shows the results of this process of design creating a marker for smartphones. On the other hand, Figure 3.12(b) shows a different marker based on virtual action taken for the camera and the algorithm. The initial test with this pattern was the acquisition of actions based on the response of the user selecting the action in the physical pattern (see Figure 4.23). The general purpose of this pattern is for the integration of wearable devices in the application.

(a) MAR application Pattern Smartphones.

(b) MAR application Pattern Smart Glasses.

Figure 3.12: Design based on examples from Vuforia Augmented Reality (VAR). (a) Design based of features for Smartphones. (b) Design based on features Smart Glasses Vuzix M100.

Finally, the creation of different patterns is made in order to be recognized by the system as different entities, these are placed around the factory floor. The position of the patterns is according to the general idea of the implementation of AVITA making a relationship between the markers and the position of the patterns around the ten cells.

Block Definition Diagram

The BDD defines a high-level structure of Augmented VIsualization for Transparent Factory (AVITA), the use of this diagram was the acquisition of the properties, operations and references of the definition of the internal components of the system. In simple terms, the BDD shows the system classification tree in the initial state of the design process. Figure 3.13 describes the first BDD with the main blocks of the system, integrating in two main blocks Hardware and Software. At the same time, these blocks define the structure of the systems and the reference of each element. The created diagram could help for future development and integration of more elements to the system.

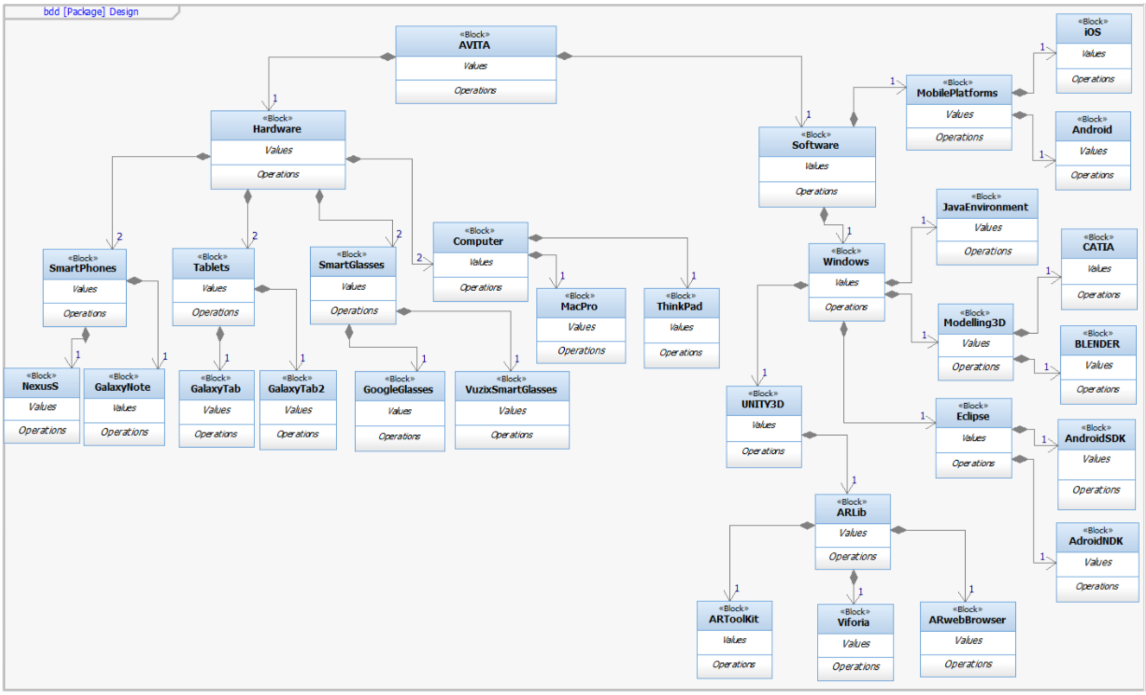


Figure 3.13: Block Definition Diagram (BDD) System Description of AVITA.

Summing up the mentioned creation of the BDD starts from the subdivision of the hardware and software. Then, the hardware block contains the different elements used to the development of AVITA, for example, in this version of the BDD general elements as Smartphones, Tablets and wearable devices. On the other hand, the software block contains elements to integrate the application such as the OS and third-party applications as UNITY3D described with more detail in Chapter 4.

After the integration of the BDD and finished the design process in the Spiral development process, new elements are found after more tests of AVITA, Figure 3.14 describes in detail the interface design and information flows within the system. In other words, the system is designed with in detail for future updates of new developers that would like to improve AVITA from the basic versions, containing new representative operation and values. For example, basic elements in the user interface, defined with more details in Section 3.3.

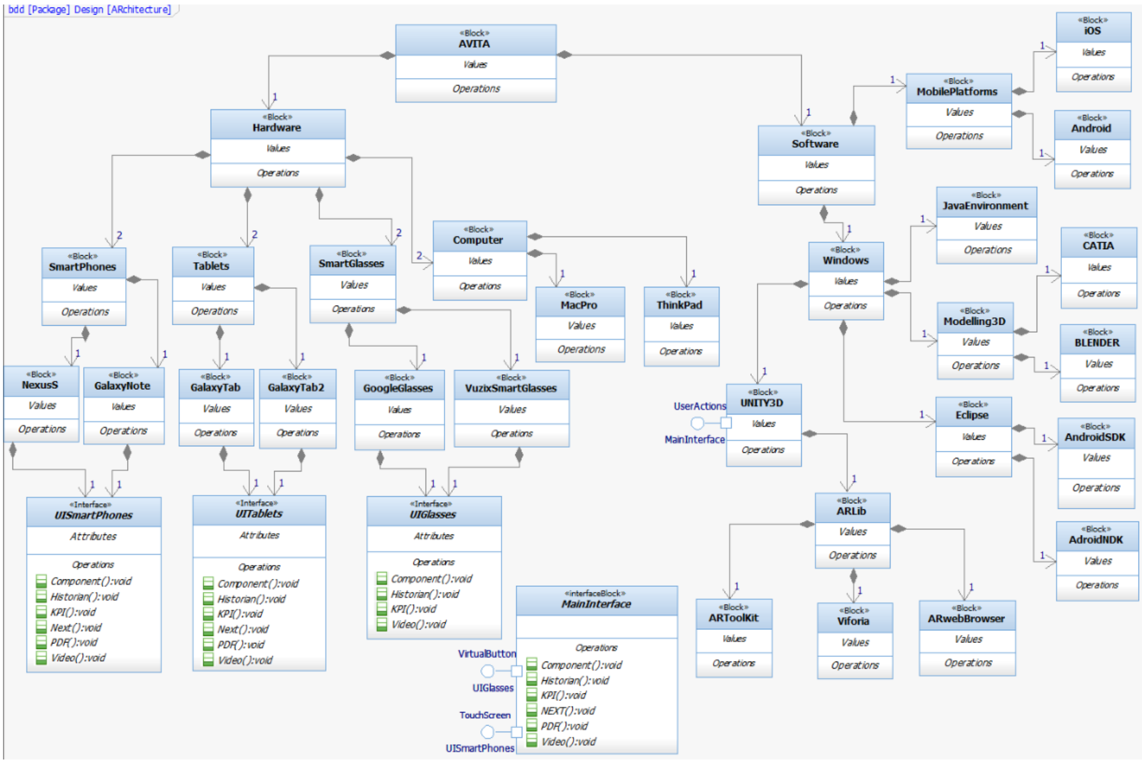


Figure 3.14: Block Definition Diagram (BDD) System Description integrations information flow and the interaction with interfaces.

The final BDD describes the connectivity within the elements, making easier the analysis of the system in order to implement the application (see Chapter 4). In this case, many devices are integrated by different interface elements, the clear example is the difference between the interface for Smartphones and wearable devices. In addition, the main interface integrates the information and variables that change according to the responses of the user.

3.3 User Interface Analysis and Design

Until now, the design process compiles information from SysML diagrams, in order to create the interfaces and the basic interaction with the users. This section describes in detail the results of using the created Sequential Diagrams that describe the interaction with the user and the application (see Appendix E). The final results in this section is the creation of the user interface for AVITA including the development of more friendly interfaces to the user. The first test of the interface was created based on the objectives mentioned in Chapter 1 and basic concepts of AR.

In this process, the main achievements were the set-up of the tools for creating the MAR application, including the display of a simple object. Thus, Figure 3.15 shows the design of the initial idea of the application before the technical process start and the following list describes the achievements in the initial stage.

- Technologies Integration (see Chapter 4).
- Augmented Robot (see Appendix A and Appendix B).
- Implementation of Tracking algorithms (see Chapter 4).
- Testing the AR marker design (see Chapter 4).

Figure 3.15: Interface Design AVITA V1.0.

Furthermore, the interaction continues with the implementation process. Then, after the successful integration of the mentioned elements in this section, the second interface was created containing more details in design, starting with the integration with programable actions. Figure 3.16 shows the new elements integration, in comparison with Figure 3.15.

Figure 3.16: Interface Design AVITA V1.1.

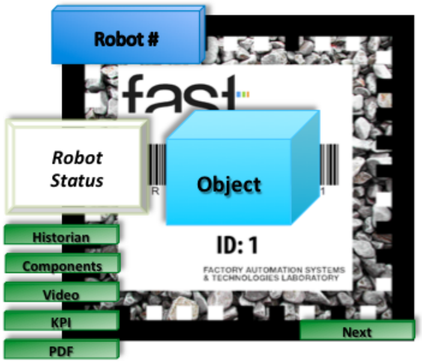
The following list describes the new elements integrated into the described interface. Also, is included the integration of more elements in the list of improvements based on the elements defined in the Section 3.2.1.

- Augmented Robot (3D models include more details and quality resolution).
  - 2D text (Adding text in instructions for the user).
  - 3D text (The name and status of the robot is created in 3D).
- Basic Animations.
  - Kinematics (Interaction with 3D environment in UNITY3D).
- User Experience.
  - Animation (Interaction with the basic GUI).

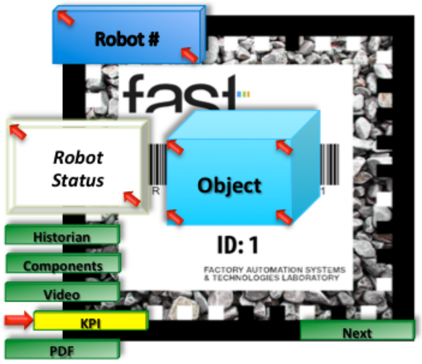
Finally, after testing the first versions of the interface, the main menu is created in order to have a better user experience using AVITA in a real scenario. In addition, more elements as animations and user interaction are included based on the compilation of information taken for previous sections as the created SysML diagrams in the design process (see Appendix E).

As a result, Figure 3.17(a) shows the general interface for AVITA after the different interactions of design with the connection to other modules of the MS. The main interface contains the elements described in Section 3.2.1 and each element has an action that the user can activate. Figure 3.17(b) shows the flow when the user performs and action. Note that in this example, the animation is activated moving the robot to the corner of the AR marker. Figure 3.17(c) shows the animation moving the KPI to the region of interest.

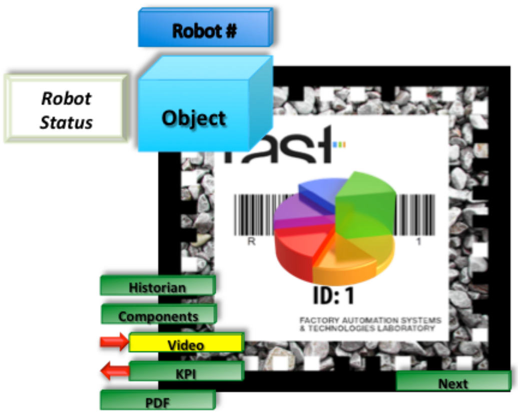
Finally, the user selects another action activating the animation of two elements, one for display a new element, another to move out of the region of interest of the user. Figure 3.17(d) shows the new interface integrating more complex elements, in this case, the video element in selected to be displayed. In addition all the responses of the user are animated showing the interactivity in AVITA.



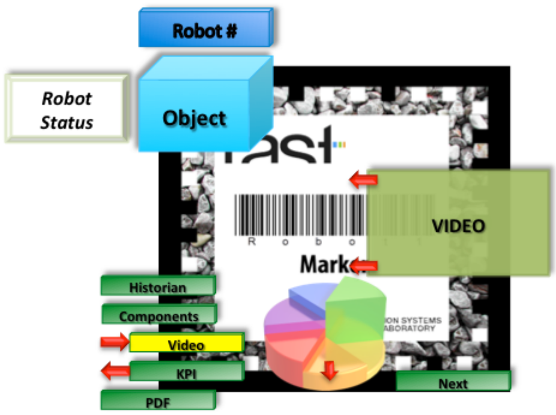
(a) AVITA main Interface



(b) Animation activated based on the user action (KPI).



(c) Selection of multiple options in the interface.



(d) Displaying new elements of interest.

Figure 3.17: User interaction with AVITA: (a) AVITA main interface. (b) Animation activated after an user action. (c) Selection of multiple options. (d) Displaying new elements of interest.

The following list includes the new elements mentioned in the design process.

- Augmented environment.
  - Robot (Textures added).
  - Components (Each component of the robot is added).
  - Lights (Illumination of the 3D environment).
  - 3D and 2D Key Performance Indicator (KPI)s, 3D Videos, PDF sheets.

3.3.1 AVITA Architecture for Manufacturing Systems

This section describes with more details the logical integration of AVITA for MS. The application was developed as a standalone application for ASTUTE project, at the first stage of the planning process the idea was simple for proving the concept of MAR for manufacturing systems, based on the current market of applications with AR and MS. Then, after several design tests with different technologies the base of AVITA was created. In a short period of time, the application was tested in mobile devices with huge success. Figure 3.18 shows the module interactions in an initial state of AVITA, the module start from the capturing of the AR marker, then the selection of the AR library with the tracking algorithms, then UNITY3D engine process the data and create the augmentation of the objects in 3D and finally the project is exported for Android devices.

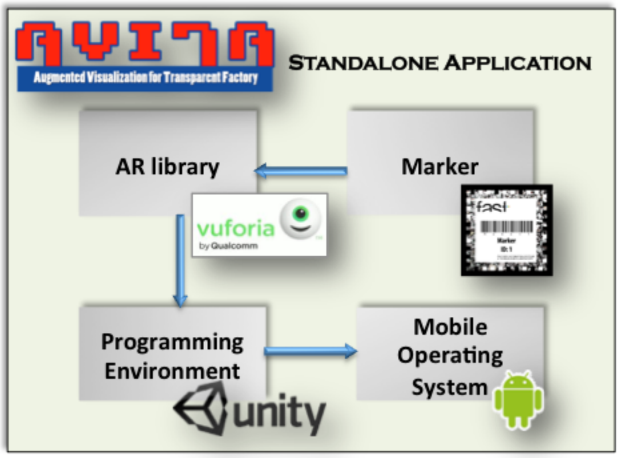


Figure 3.18: Modular Description of AVITA.

Finally, AVITA contains the main concepts of MAR including the compilation in a mobile operating systems. In addition, at this point of design some tools were tested in order to select the correct library and programming environment for the AR application.

Integration with other Platforms

After the first implementation of AVITA, some modules were included in order to establish the communication with the factory floor. As was mentioned along this Chapter 3 based on many requirements and SysML diagrams, the system was designed to achieve the objectives mentioned in Chapter 1. Figure 3.19 describes the logical communication with the other external modules. For example, the integration with ASTUTE project.

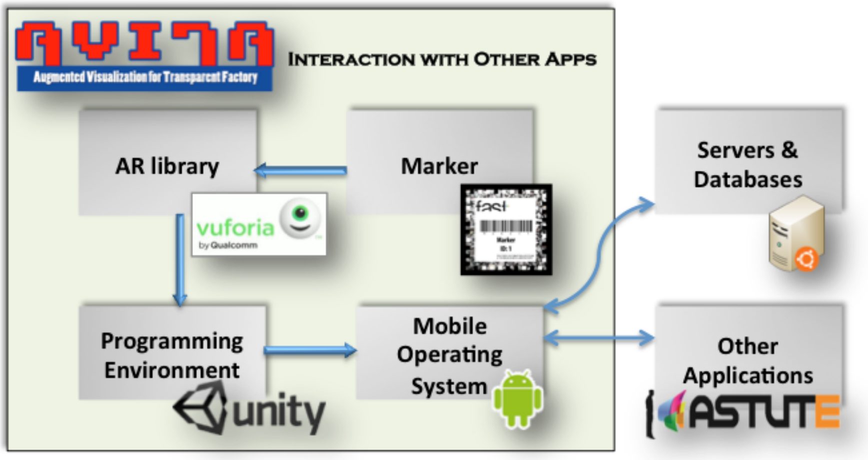


Figure 3.19: Modular Interaction Description.

The document [NLLC13] describes in detail the integration with AOS and web technologies. This example is taken into account to integrate the described solution (AVITA) with other applications. After the last implementation of AVITA, this integration was considered as an option to connect and integrate AVITA as a complementary application of the main architecture of ASTUTE. As a result, Figure 3.20 in created, describing the main architecture of the technical implementation of the creation of a builder for Adaptable HMI for Mobile Devices. This document includes the general concept of the integration of AOS with web applications in order to generate adaptive interfaces. At the same time, an industrial environment in taken as a use case for this implementation. In conclusion, the technical implementation and results of this work can be adapted to AVITA in order to receive and send information through the MS.

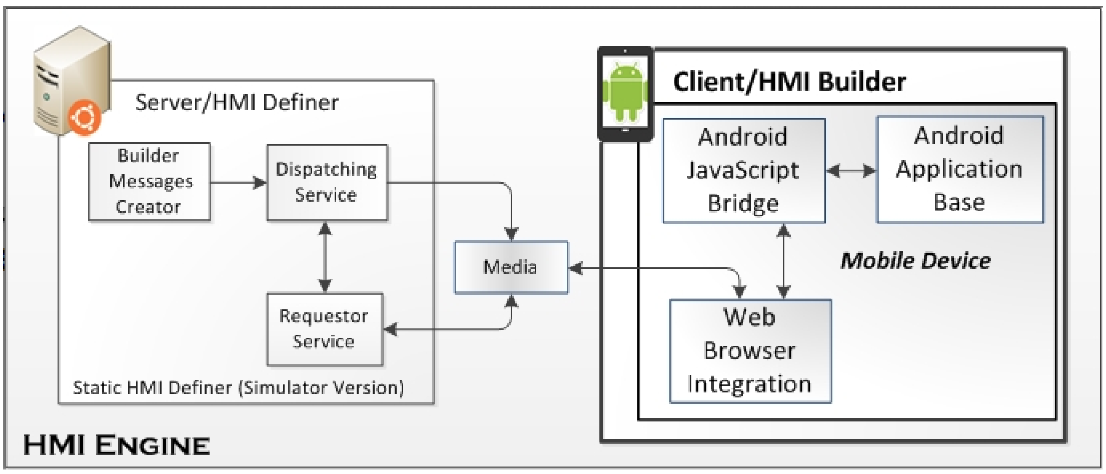


Figure 3.20: HMI Engine ASTUTE [NLLC13].

4. IMPLEMENTATION

The application was developed in Tampere University of Technology with the objective of integrating a MAR application into a real scenario for FA. The proposed scenario is taking place inside a factory floor applied for monitoring MS taking care of possible events and situations in the production line. The users with the role of monitoring and controlling the production line are able to open and install the MAR application (AVITA).

The patterns are collected and placed in each cell of the factory floor. Each cell has different purposes and every cell is continuously working. The users can in anytime see the process using the MAR application just placing the device in front of the target. After the user has the device in position different animations start running taking information from the MS, for example, the user is able to see the 3D robot kinematics in the main page of the MAR application. On the other hand, the MAR application integrates a HMI in order to interact with the user displaying basic action to the different users across the factory floor. In addition, the user can switch between elements in the application delivering a real experience with AR showing information related the production line. The objective of the implementation of the MAR application is safe time and resources using the application, showing in runtime information concerning the users involved in the factory floor.

This section describes in detail the creation and compilation of a MAR application. After, reading this section is easy to see some gaps in the implementation part. For this reason, is recommended to follow the guide described in the Appendix sections A, B, C and D. The current implementation in contrast with the mentioned sections covers the installation of the VAR library with the compilation of a simple project example, the creation of the AR camera, the integration of the trackable image, and testing mode.

4.1 Application Description

The MAR application was developed for mobile devices and wearable devices, in particular, for AOS. The UNITY3D is used to generate 3D elements and integrate the application with Android devices. Also, the MAR application is adapted to an IDE for communication and integration purposes with other applications [NLLC13]. The library VAR [Qua12] is implemented as an extension of UNITY3D, containing the concept elements of AR, such as capturing and register based on tacking algorithms.

AVITA displays 3D information for a factory production line, where the user can have a better perception of the current events during the production (control, management or maintenance). AS it has been previously mentioned, the MAR application is developed in three different versions. As a result, Table 4.1 shows the main functionalities of each version including the design elements presented in Chapter 3, the render quality of the 3D objects, the communication module implemented and the tracking algorithm implemented. The application shows the 3D objects overlapping the trackable objects detected by the camera, displaying information related with layout of the factory. Then, based on the analysis and design of the user interface presented in Chapter 3, the user can navigate through the elements of the application selecting in the interface which elements are relevant to the current task of the user.

The elements displayed by AVITA are generated based on the distance and orientation of the user's device also in the position of the target (Trackable Object or AR marker). The device can move along the target to inspect in detail the displayed 3D information. The animation is created based on the user's actions also in the system current events. Moreover, the user can switch the visualization using the interface (Chapter 3).

Table 4.1: Augmented VIsualization for Transparent Factory (AVITA) Application Versions

Version	Render Quality	Communication Module	Virtual Tracking
Smartphones	Medium	Included	Not Implemented
Tablets	High	Included	Not Implemented
SmartDevices	Medium	Not Included	Implemented

4.2 Equipment and Material

The elements that integrate the MAR application are described in the following section, dividing the section into two main blocks, this two subsections follow the structure given by the requirement diagrams and the Block Definition Diagram (BDD) described in Chapter 3.

4.2.1 Software

Graphic Engine and AR Library

The selected graphic engine is UNITY3D. Cross-Platform Game Engine (UNITY3D) is the software tool where the 3D objects and the factory 3D layout is created and the AR library is integrated in order to use the tracking algorithms for AR. The Qualcomm VAR is integrated as an extension of UNITY3D to prove the basic concept of AR. However, is was mentioned in Chapter 2 this is not the only option on the market to develop a Mobile Augmented Reality (MAR) application, but based on the requirements of the system this set of technologies satisfy the list of requirements mentioned in Chapter 3 especially in the SysML requirement diagrams. On the other hand, the VAR is a flexible solution, which integrates external sources of applications to create, animate, integrate, deploy and testing applications to mobile devices including Android OS and iOS. For this reason, UNITY3D will be used as an IDE integrating the AR library with the layout of the factory in order to build AVITA.

Operation System and Requirements

The MAR application contains a specific set up of technologies. Firstly, to start the integration of technologies is required the selection of an OS, for this implementation the base of the programming environment is Windows Operating Systems (WOS). As it is mentioned in previous chapters, the application is developed for mobile devices targeting in the first test phase the implemented and integration for AOS. On the other side, UNITY3D and the complementary elements of the MAR application such as VAR and Eclipse IDE are also available for MacOS and Linux including the deployment and development of the entire application for iOS. Figure 4.1 shows a general graphic representation of the elements that integrate AVITA. In addition, the following list represents the entire set up of the elements mentioned including a description of the requirements to complete the installation and to development on the MAR application.

Figure 4.1: Technologies Integration for MAR applications

1. **Windows PC:** The operating system minimum requirement is Windows XP or higher.

2. **Java Development Kit:** Java SE Development Kit SE 7u1, available in the oracle main web page [Cor13].

3. **Cygwin Environment:** UNIX environment command line interface for WOS. This interface compiles files created in C and C++, for this implementation libraries for AR. The minimum requirement is the version 1.7.9-1 and it is not needed using MacOS or Linux [Sol95].

4. **Eclipse IDE:** Eclipse Indigo or higher [Fun12]. The IDE is needed in order to integrate the entire solution [NLLC13]. On the other hand, the plugins for AOS should be implemented following instructions from the Android web page [All08] or from Eclipse market place. The following list describes the installed versions in the environment, currently this can be avoided from other easier sources [All08].

   - **Android SDK Downloader:** version Tools revision 16 [All08].
   - **Android Developer Tools (ADT):** version SDK revision 16 [All08].
   - **Android SDK:** Platform support [All08].

5. **Android Native Developer Kit (ANDK):** is a toolset that allows the implementation of native code languages such as C and C++. The minimum version is ANDK 7 available in the main Android web page [All09].

6. **CAD/CAM tools:** The CAD/CAM tools are used to design and create basic 3D elements. This 3D elements define the Layout of the environments . For instance, Computer Aided Three-dimentional Interactive Application (CATIA) was used to design some elements of the factory such as the robot and theirs components (see Appendix A).

7. **Vuforia Augmented Reality:** The Vuforia platform enables Augmented Reality application experiences. Vuforia is a software platform that uses top-notch, consistent, and technically resourceful computer vision-based image recognition and offers the widest set of features and capabilities. The platform integrates a UNITY3D extension adaptable to any project created in UNITY3D [Tec05]. The following Figure 4.2 shows the VAR platform describing the elements containing inside the VAR SDK [Qua12].

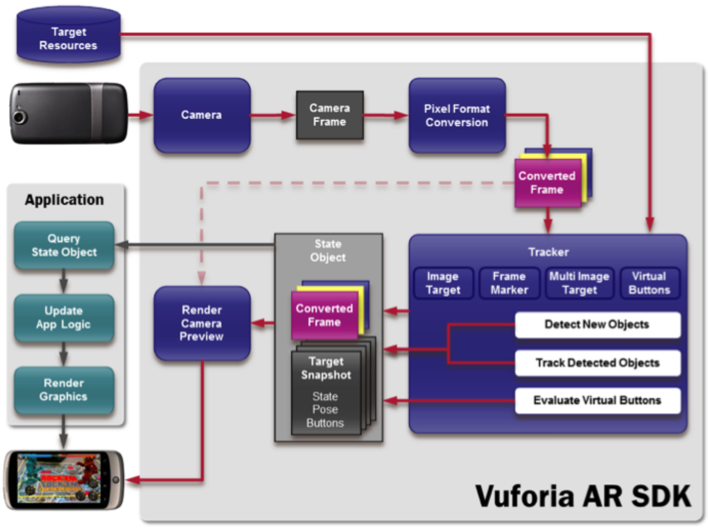


Figure 4.2: Vuforia Augmented Reality SDK [Qua12].

8. **3D Computer Graphics Tools:** These tools are described in the following list. Appendix A describes with more details the use of these tools in order to create or generate 3D environments. In more specific cases the integration of VAR.

   - **Blender:** This tool is free and has open source community support. This 3D modeling tool was used to export 3D elements to UNITY3D. The use of this tool is not described in detail in this document. In addition, Appendix A makes a reference to this tool.

- **UNITY3D:** It is a graphic engine able to compile and import the generated 3D objects from other sources such as Blender or CATIA. In addition, integrates an easy creation of animations and plugins with other tools making easier the development of the applications. The main advantage of this tool in the implementation of a MAR application is the capability of integration with third-party applications such as VAR combining the source code from C and C++ libraries. Finally, UNITY3D includes more solutions to deploy the developed application to different platforms including WOS, MacOS, Linux, iOS and AOS. Thus, the current development can be easily emigrated to other platform without any problem.

Software Integration

The integration of the software is one of the main objectives of this document. In order to create a MAR application a set of requirements are needed also mentioned across this document. At this point of the implementation phase, several technologies are chosen. Figure 4.3 shows the set of layers used in order to create an AR application in UNITY3D. The Unity Engine is the heart of the application and is the component used in to render the 3D environment and is the connector with the real hardware. Then, the Vuforia SDK contains callbacks for events when a new camera image is available, a high-level access to hardware units and multiple trackables (tracking types). For example, Image target, Multi-Target, Frame Markers and the real-world interactions (Virtual buttons).

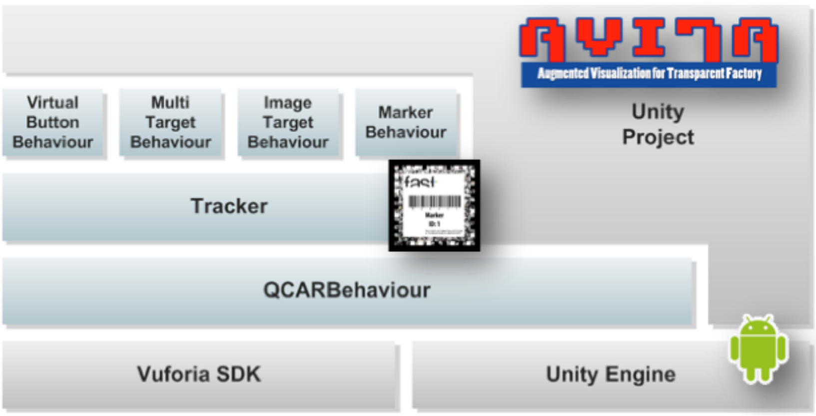


Figure 4.3: Vuforia Augmented Reality Integration with UNITY3D [Qua12].

Along the implementation phase, several interaction are made with the Spiral methodology described in Chapter 3, since the initial version of AVITA until the development based on a Unity project example including the creation of some of the 3D elements, the programed interaction between user-application and the design of the AR patterns.

4.2.2 Hardware

This section contains the list of the used hardware to integrate and test the MAR application. The list was created based on results from Chapter 3 especially using the BDD diagrams.

Smartphones

The smartphone was included in the list of devices due to their portability and the fact that nowadays everyone has one mobile device all the time. In several interactions of implementation of AVITA was used Nexux S.

1. **Processor:** 1 GHz Cortex-A8.
2. **Display:** Super AMOLED capacitive touchscreen, 480 x 800 pixels, 4.0 inches (233 ppi pixel density).
3. **Network:** HSDPA 900 / 1700 / 2100.
4. **Camera:** Main (rear) 5 Megapixel, 2560 x 1920 pixels,.
5. **AOS:** Android 2.4 upgradable to v4.2 (Jelly Bean)

Figure 4.4: Nexus S model GT-I9020.

AVITA is adapted to the size of the screen automatically based on the instructions to deploy an application for mobile devices in UNITY3D (see Appendix D). The interface change according to the size of the screen. In other word, this version includes a small interface and medium resolution 3D image. The advantage using this device is that the user can carry the device all the time due the size of the smartphone.

Tablets

The tablet used to implement one of the initial approaches was the Galaxy Tab 2 with Android 4.1. This tablet is one of the first tablets with a wide screen and dual processor with grand acceptance in the mobile device market. The following list describes the main features of the device.

1. **Processor:** 1GHz Dual-Core Processor.

2. **Display:** 10.1” WXGA (1280x800) PLS TFT.

3. **Network:** HSPA+21Mbps 8850/900/1900/2100.

4. **Camera:** Main (rear) 3 Megapixel.

5. **AOS:** Android 4.0 (Ice Cream Sandwich)

Figure 4.5: Galaxy Tab 2 10.1".

AVITA in the initial state of development was tested in the Galaxy Tab, in order to see the elements with more details and resolution. The application installed in this device contains full functionalities including resolution and animation due to the specification of the device, AVITA was successfully unpacked and tested in this device. On the other hand, the drawback of this device is the size and for some user could be uncomfortable to use when a bad situation occurs and is taken into consideration for future usability tests.

In general, the final version of AVITA is tested for tablet devices and included in the initial usability test. In the same way, the documentation and implementation related the device is also included in ASTUTE documentation. Currently, the development is moving towards to new challenges implementing AVITA for wearable technologies.

Smart Glasses

The smart glasses are a new branch of wearable smart devices containing similar elements containing into a smartphone. This type of devices will be taken into account in the requirements of design to test the different component proving how hard is the implementation and adaptation of AVITA in these devices.

This hardware is now available in some markets and other are still in process of development. The example used in this scenario is the smart glasses developed by Vuzix a company with huge experience in the domain of AR and hardware development in particular for smart glasses technologies. The device is selected based on a prototype development by Vuzix (Smart Glass Device M-100 Hands free Smartphone Display). The following list describes in detail the main specifications and features of the smart glasses used in this implementation.

1. **Optics Engine:** Display resolution WQVGA 16x9 displays.
2. **Processor:** OMAP 4 Processor (Fully optimized Android 4.01).
3. **Display:** 10.1” WXGA (1280x800) PLS TFT.
4. **Camera:** HD 1080p.
5. **Head tracker and GPS:** 3-degree of freedom head tracking, Compass and GPS
6. **Integrated Battery:** Up to 8 hours
7. **Operating Systems (OS):** Android / iOS.

Figure 4.6: Vuzix Smart Glasses M100 [Vuz12].

At the same time, AVITA was developed for this device integrating new functional and non-functional requirements to the system (see Chapter 3). Based on the features of the device different option to implement AVITA in wearable devices appears. Figure 4.7 shows the different modes of operation. This means that the application development for this type of devices can lead to the complete installation of the application using the Native compilation running the application on the M100, using the device as a HMD installing the application full in the mobile phone using bluetooth to communicate both devices or install the application in both sides. The Smart Glasses Vuzix M100 has a huge impact on the mobile market due to the innovation and development by high standards and the adaptability with more technologies.

Figure 4.7: Vuzix Smart Glasses M100 Types of Modalities [Vuz12].

The Vuzix Smart Glasses M100 contains a dedicated processor running the Android v4.0 operating system as its core OS (otherwise known as Ice Cream Sandwich version 14 of the API). The M100 is designed to operate independently or work in concert with a smart phone, using wireless connections between the devices to manage everything from the user interface to applications. Over the course of the next few months, physical and functional refinements to the M100 will become available and will be reflected in the subsequent releases within the SDK. Below is a brief description of the physical features of the M100. Please refer to the specifications of the M100 for more detailed information.

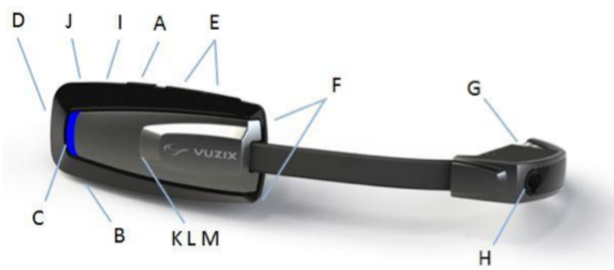


Figure 4.8: Vuzix Smart Glasses M100 Detail Description.

(A) Select "Soft Key": Physical button whose function changes depending upon the application context. It is typically used as the button to select and start applications or when selecting functions within a running application.

(B) Power Button (On/Off/Sleep): Physical button tied to the power management function of the device. Press and hold for 5 seconds to turn the device on. Press and release the button to place the device in and out of sleep mode. Press and hold for 5 seconds to display Power menu (Off/Cancel). Press and hold for 10 seconds to force power off.

(C) LED Lights (Blue and white): dependent control of Blue and White LEDs.

(D) Micro USB: Connector for charging, controlling, and providing system upgrades. Also used for extended use battery option.

(E) Navigation "Soft Keys": 2 Physical buttons whose functions change depending upon the application context. Typically used for volume control (up/down). Can be selected together and used for alternative purposes when applications are running.

(F) Microphone: 2 noise cancelling microphones.

(G) Display: WQVGA color display with 428x240 resolution, 16:9 aspect ratio. The Virtual Screen Size is equivalent to a 4 inch screen 14 inches away from the eye.

(H) Camera: 1080P, QSXGA resolution, 16:9 aspect ratio. Video recording capable.

(I) MicroSD Slot: MicroSD card provides user accessible memory.

(J) Earpad: Noise suppression audio speaker with interchangeable headset accessories. Earhook, over the head, behind the head.

(K) Connectivity: Wifi802.11,BT

(L) Head Tracker/IMU: Integrated sensors, 3 axis gyros, 3 axis accelerometer, 3 axis magnetic field.

(M) GPS: Built in GPS.

The implementation of AVITA for wearable devices is added in Section 4.4 describing the user interaction with AVITA in the different versions of development.

4.3 Architecture Implementation for Manufacturing Systems

This section describes the physical connection in order to test AVITA's compatibility with other MS. Figure 4.9 shows the general view of connections between AVITA and the MS. Based on information provided by the Spiral methodology and theirs interactions (see Chapter 3), the coding and debugging process starts. In general, this phase can be also called implementation and debugging process. This section describes in technical detail issues with the creation of AVITA including references to implementations used in order to establish communication with the different models of the entire solution.

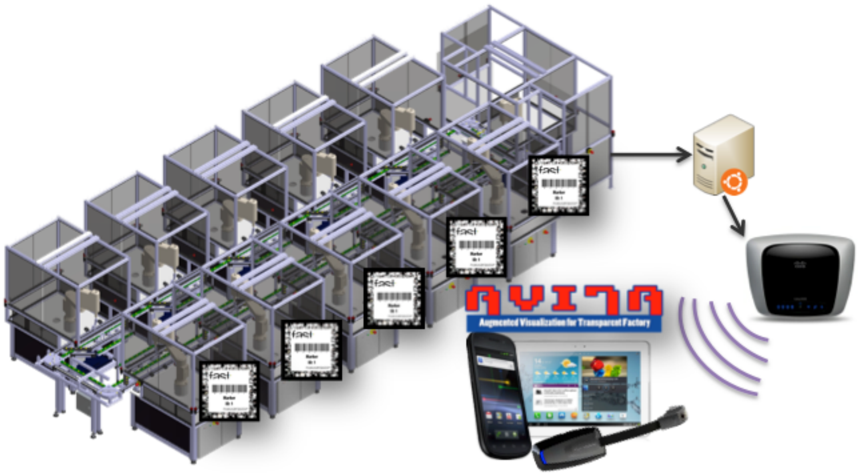


Figure 4.9: General Connection Physical View.

Finally, the different developments of AVITA tare followed based on the description of the different mentioned modules that integrate the MAR application including the communication module and integration with third-party applications.

4.3.1 Communication Modules

This section was created after the first delivery of the implementation of AVITA. After, new requirements are included in order to integrate the communication with third-party applications based on the results of design of Chapter 3, in particular Section 3.3.1 describing the integration with other platforms [NLLC13]. As a result, taking this approach to communicate AVITA with the MS.

Server Communication

The server communication channel was one of the first challenges to achieve in this process of implementation. As a result, a physical connection was created with several modules.

The communication with the server was created in order to simulate events from the MS (see Chapter 3). In addition, to this implementation phase some layers of AVITA are included in modules from the architecture [NLLC13]. Figure 4.10 is created based on the architecture mentioned in Chapter 3 supporting the new elements. The general module of AVITA is integrated in this architecture to receive the information from the MS.

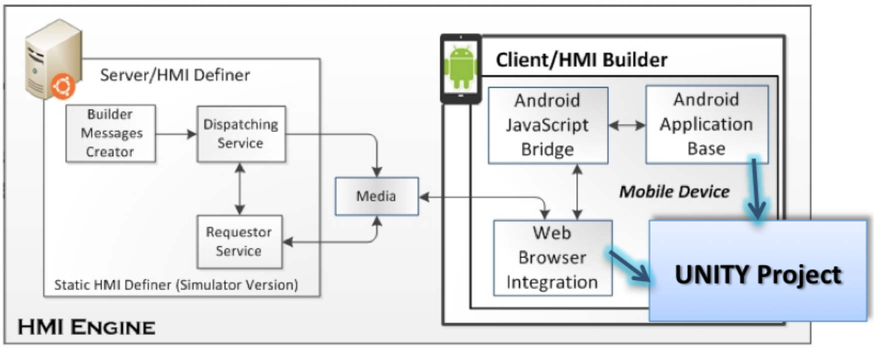


Figure 4.10: General Module of AVITA Integration with third-party Applications, Image Modified from [NLLC13].

The server is a PC capable of providing information for any application inside the factory floor. Firstly, the server is able to provide information receiving and sending java objects to the applications subscribed or connected to the server. The server simulates the behavior of a manufacturing line. Thus, the application is capable of receiving the messages from the server and react based on the events in the MS.

To achieve the communication, the application should be able to establish communication with the server, in order to display real information in runtime. Figure 4.11 shows the elements programmed in the server and the client side [NLLC13]. These methods are created to receive information from the server using android devices and web technologies especially asynchronous connectivity to receive the information at any time.

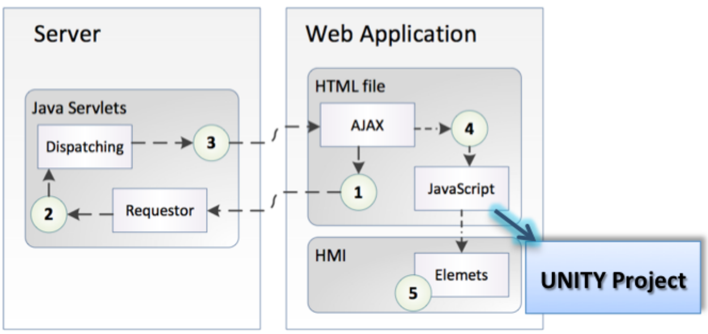


Figure 4.11: Communication Modules to Connect AVITA [NLLC13].

UNITY3D-Eclipse Integration

After the development of the MAR application, one of the biggest challenges is the creation of a communication channel to the server, as mentioned in Section 4.3.1 of development. The integration with UNITY3D is essential due that the entire MAR application is developed in UNITY3D.

Currently, UNITY3D is supported by different programming languages and many companies developing additional plugins for the development of UNITY3D, but still it is not capable for establishing a connection with a server using asynchronous communication. In fact, the set up of the project created in UNITY3D inside other environment in order to achieve the mentioned communication channel it is required. Now, with the experience achieved during the design and implementation process, it is easier the identification of the correct IDE that will be helping the emigration of the created project in UNITY3D. Then, recalling Section 4.2.1 that includes the technology integration, combining Eclipse IDE and Android libraries, this task can be completed straight forward with following different manuals from the main developer web page of Android and UNITY3D.

UNITY3D is capable of building and compiling projects inside different platforms, deploying and testing the packages installed (see Appendix D. The integration starts with the extraction of the files created in UNITY3D engine and restoring those files into Eclipse IDE. The purpose of import the files into Eclipse is to acquire more control and give more functionality to the MAR application. For example, the communication with the sever and devices. As is illustrated in the Figures 4.10 and 4.11, UNITY3D generates some files especially for the operating system compiled. Thus, those files can be exported as a library also as a project in Eclipse allowing to see the source code of the application in Android. In the end, the application can be running inside Eclipse, showing the UNITY3D application inside the Android engine.

4.4 Application User's Interactions

The results of the architecture and design of the user interactions, pre-defined in Chapter 3 are taken and implemented in this section. The result of the design process was the detail generation and integration of the user interfaces, from basic form until complicated interactions with the HMI. The initial user interaction in AVITA is shown in Figure 4.12. The user hold the device before the application is activated. In addition, the user starts the MAR application showed in the device main screen, for this case AVITA icon. Then, the MAR application loads some elements such as the AR library and camera algorithms. Finally, the user can hold the device and track the pattern until the application detects the pattern and the 3D object is displayed.

Figure 4.12: Initial User Interactions with Mobile Augmented Reality (MAR).

Afterwards, the application starts and the user is able to interact with the virtual content displayed. The interactions are created based on the use case diagrams in detail with the sequential diagrams described in Chapter 3. Figure 4.13 shows the 3D created environment after the design process, these interactions are made based on new requirements of the systems, basically are created in the same scene to avoid problems of delays when the user interacts with the MAR application.

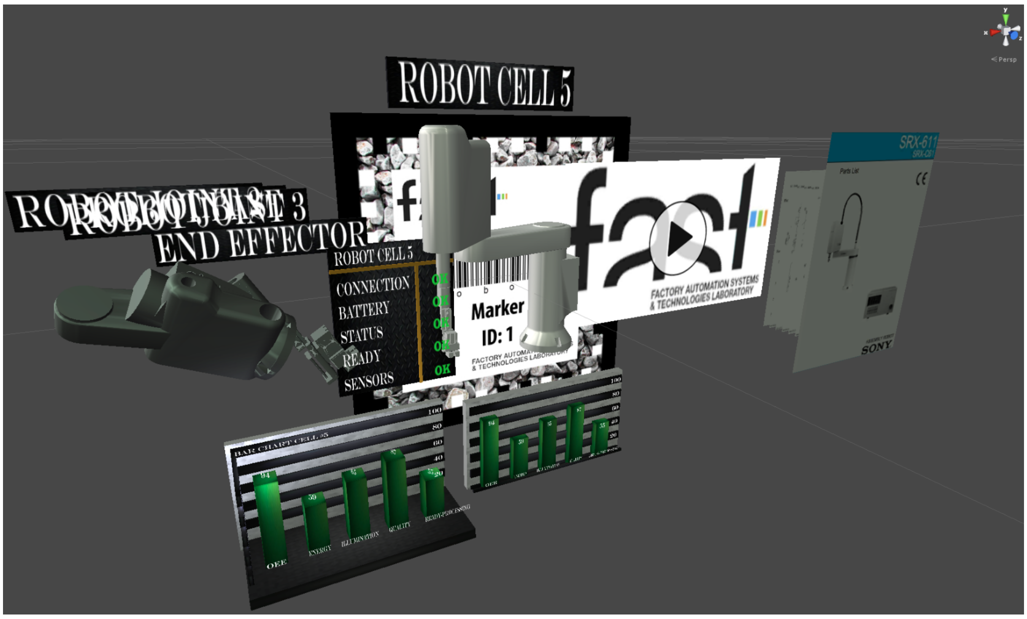
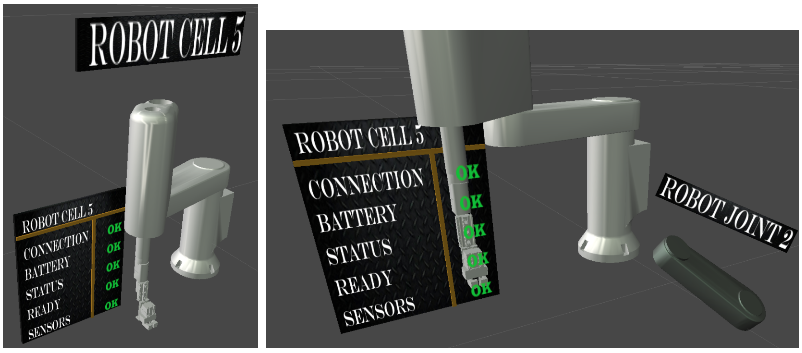


Figure 4.13: Interactions with Virtual Content in UNITY3D.

The interactions are defined by the user interface and the action of the user, as is defined in the design process. In the beginning, the animation is created based on the possible actions of the user. For example, if the user starts the application and after a while decides to change the visualization of the application, changing from the main view to a component of the robot.

Figure 4.14(a) is the main screen of the animation mode in UNITY3D after the user selects the interface the option, in this example, the component element activates the animation in UNITY3D moving the robot and information to the corner of the AR pattern. Then, one of the components is displayed in front of the user region of interest, making easy the inspection of that particular component. Figure 4.14(b) shows the transitions between the action of the user and the animation.



(a) Main View Interaction. (b) Component Switching View Interaction.

Figure 4.14: Images Taken from Implementation Process in UNITY3D (IDE). (a) UNITY3D Main View Interaction. (b) UNITY3D Component Switching View Interaction.

Summing up, this section is used to translate the result of the design process to real implementation in the selected platform. The rest of the interactions are described in detail in Chapter 3 describing the design of the HMI.

4.4.1 User Interface Development

The created user interfaces for AVITA are divided in by type of the version and device used in the process. This section describes the development process of the created HMI along the development of AVITA. The interfaces are defined in detail based on the results of previous interaction with the Spiral development methodology presented in Chapter 3. As a result, the following subsection describes the issues found after each interaction with the development process.

User Interface AVITA V1.0

The initial version of the interface in the implement process starts with the prove of concept of AVITA, this means the integration of basic elements integrating AR in the process. The user experience is basically none due to absence of the interface and interaction in the application. However, following the guide included in this document it is possible to reach this point of implementation reading carefully the attached appendixes A, B, C, D.

Until now, UNITY3D compile the Android libraries including the project containing the AR libraries. As a result, UNITY3D creates and icon ready to be executed and tested. Then, the pattern is in position and the tablet is targeting the marker. Thus, the application is ready to start tracking. Figure 4.15 illustrates how the application is running showing the initial version of the MAR application.

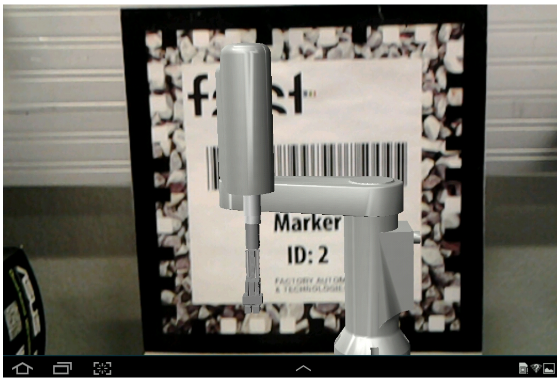


Figure 4.15: Initial development Testing MAR applications.

User Interface AVITA V1.1

After some improvements the application look different including more 3D elements. Figure 4.16 shows the MAR application with the new requirements of information related the object displayed. However, this change in a short period of time due the internal interaction of the Spiral development methodology. In addition to this interface the application has animation included based on the kinematics of the robot, creating a simulation of real functionality of the robot in the production line, also a simple button is created to interact with the user switching the between scenes in the MAR.

Figure 4.16: AVITA Robot Kinematics.

The AVITA development process starts with the selection of the distribution of scenes in UNITY3D with the general idea of the integration of more modules and scenes based on future developments. However, the result of that process was the delivery of a slow application, due to the loading of several scenes and animations that integrate the user interactions with the virtual content. After, the run-time of the scene in UNITY3D the application starts generating delays in the navigation, waiting 3 or 5 seconds to see the next animation.

Meanwhile, this test integrates basic user experience and interaction with the user. Afterwards, new 3D elements are integrated into this version containing rotations and translation of the object around the region of interest in this case, the origin of the AR pattern. The robot kinematics was improved, the animation is included and programmed in JavaScript and C-sharp. In conclusion, based on the design process more scenes are generated containing the different robot components in order to achieve the requirements of the system. The following list describes in detail the elements included in this process especially in Figure 4.17.

1. **Main scene:** This scene contains the robot kinematics displaying the robot. The main scene is a basic HMI that starts the animation and displays the information retrieved from the MS (Kinematics), also creates the connection between the different scenes.

2. **Robot Base:** This scene is activated once the user presses the action button, displaying the base component of the robot. In addition, the animation of this component is a simple rotation along the "z" axis.

3. **Joint 2 and 3:** These components are attached to the robot base, in order to create the robot's kinematic. As is defined, in the step 2, this two components are also animated along "z" axis.

4. **Gripper:** This is the last scene where the gripper can be inspected, including the same animation and the link to the main interface.

The user starts with the sequential actions given by the provided interface. In this version of the user interface, the objective was the creation of a basic user interaction based on a loop selection, displaying the different parts of the robots. these interactions are described in detail in Chapter 3.

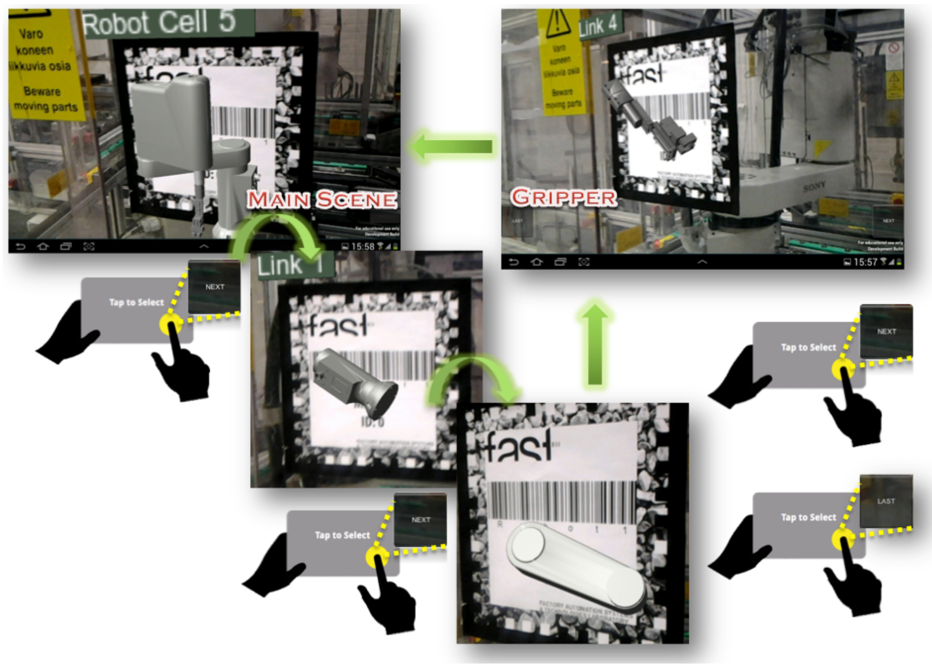


Figure 4.17: User experience in AVITA Version 1.0.

At the same, more features are collected from the AR library. The library includes functionalities based on the device position, this means that the device can be moved across the pattern and can be used in different positions, distances and angles. Figure 4.18 represents the robot seen in another angle until the camera can see the trackable image the robot is still in the screen and is possible to zoom in and zoom out depending on the user position. The application has controllers where the users can switch between components of the robot. After, the user selects the action the scene change showing the base of the robot as a first component in the list. This animation was made in order to provide different views to the users when they want to inspect a specific component of the production line.

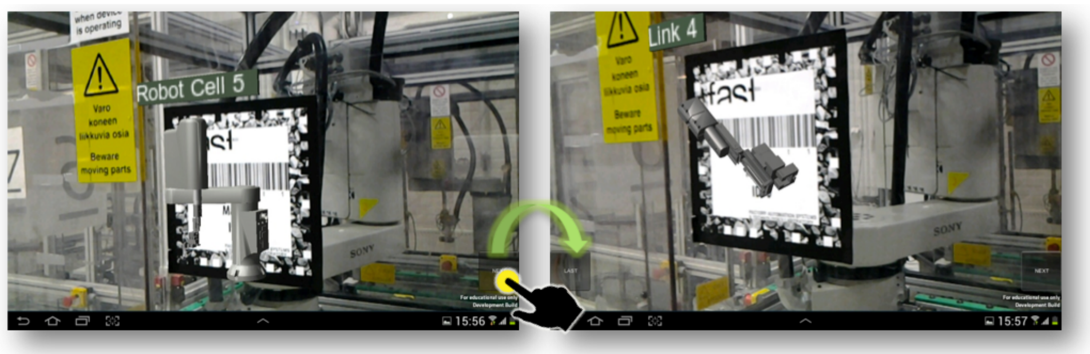


Figure 4.18: AVITA HMI Interaction with the User.

User Interface AVITA V1.2

The development for the interface version 1.2 is based on the design process results (see Chapter 3). The interface is programmed and created in UNITY3D with the objective of improve previous versions and trying to avoid founded issues in initial tests. For example, the generated delay between the user command and the application response.

The solution of many problems in the initial test of AVITA was the development of the main scene containing a complete set of content, in order to turn on/off the 3D components based on the current state of the system, avoiding performance issues, as a matter of fact, this main scene solves the performance issue in AVITA. In addition, to the current solution, a connection with the VAR is made in order to create the different animations when the user selects one option on the interface. Figure 4.19 shows the interfaces from the main screen until the transition between two elements is made.

The user starts the selection of the component option on the interface, animating the robot to the corner of the pattern and displaying the first 3D component as is defined in Chapter 3, this animation is predefined, smooth and fast. Then, the user selects the video option interface, displaying a video about the current production and functionalities of the robot. Finally, the user can change the selection at any time, last screen shows the transition to the KPI option, moving the video elements to the original position and displaying the new graph component in the region of interest including the current state of the system.

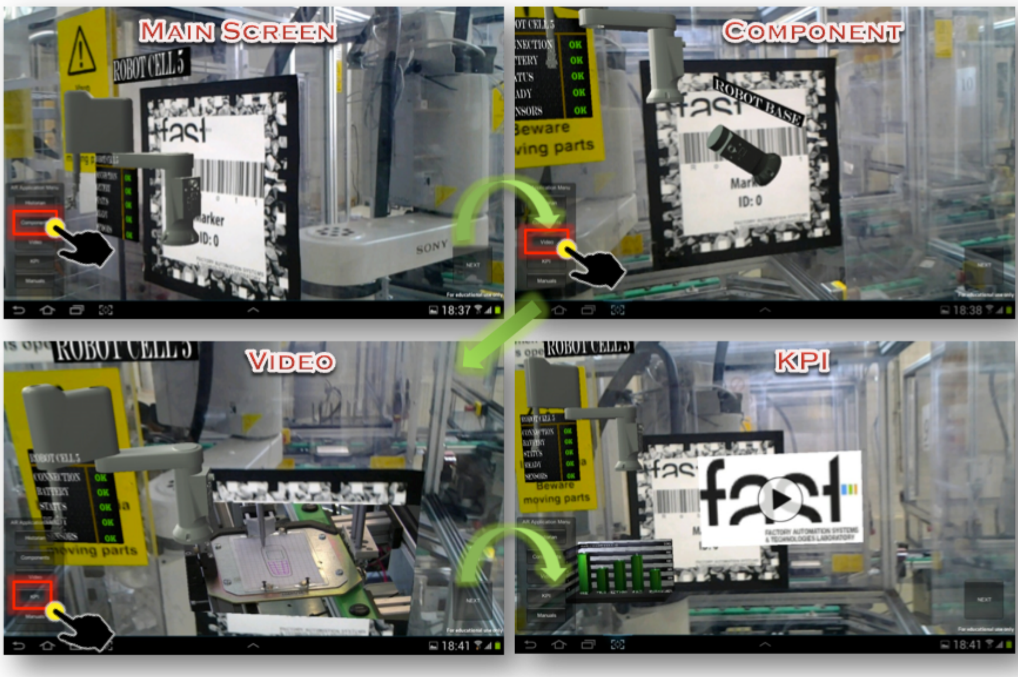


Figure 4.19: AVITA new Interface Interaction.

The user has different interface options to inspects in detail the information related the MS. First, the user can move the device along the pattern watching the 3D content in different angles. Then, UNITY3D has the capability to render the elements based on the position of the user with a huge quality resolution. This means, that basically the zoom in AVITA is programmed based on the distance between the pattern and the camera. Figure 4.20 shows the application working in closer look targeting the AR marker, the KPI is selected and displayed showing the information provided by the MS . This example is clear showing how the user can see in detail the actual values of that specific cell in the production line. In addition to the different option provided in AVITA, Figure 4.20 shows an example based on the scenario, if there is a problem with the robot and the reparation is required, the user can select the manual option that displays the data sheet of the equipment making easier the installation of the new part using the zoom based on the position of the device this information can be inspected in deep detail.

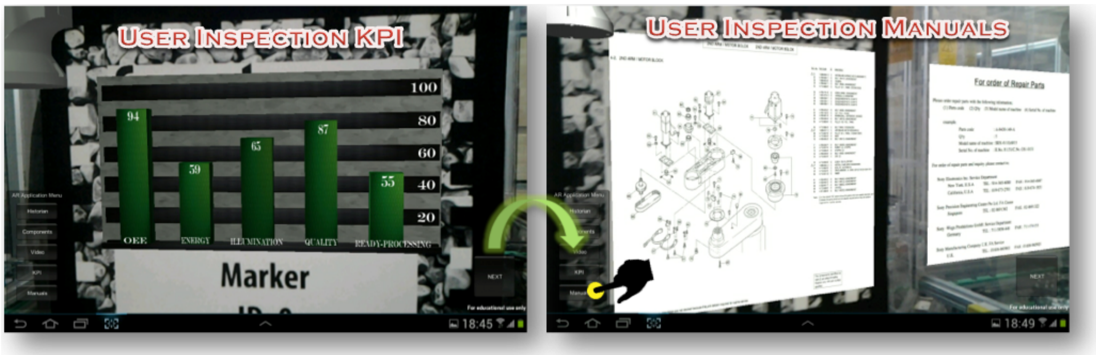


Figure 4.20: Virtual Content Closer Inspection.

Wearable Devices Interface

This interface is created based on the same methodology described in Chapter 3. Thus, the planning and analysis start again based on the new hardware requirements. Also, new parameters in the design process are required in order to create the interfaces for wearables devices. First, the interfaces are created based on the concept of wearable devices where the interface is not present or non contact with the user to an interaction. However, AVITA in this version integrates the position of created virtual objects at the bottom of the AR pattern. At the same time, the pattern is designed based on the rules mentioned in the design process (see Figure 3.12(b)). This pattern has the same elements of the other type of AR patterns using the algorithms for tracking and position. The main difference lies in the detection of the user action in the real pattern, in other words, if there is an obstruction on the image the action is triggered. Figure 4.21 shows AVITA main interface in the version for wearable devices.

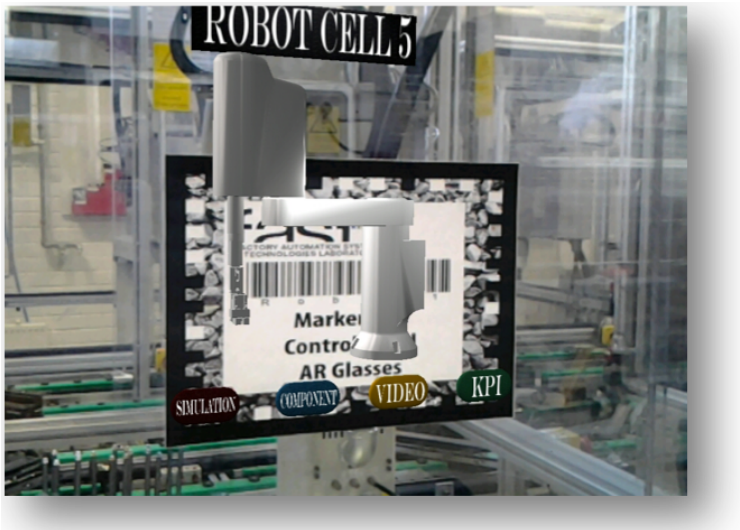


Figure 4.21: AVITA Main Interface for Wearable Devices.

To develop an application for this type of devices it is needed the specific SDK version in order to test the functionalities of the software after the installation in real devices. The Vuzix smart glasses M100 contains their own SDK and compiled libraries for Android devices. This software can be installed to emulate the device in the IDE and test the application as an Android application. Figure 4.22 shows the emulation running in Eclipse IDE. Then, AVITA can be exported in this platform to test if the MAR application will run without any problem in the platform after be installed in the real device.

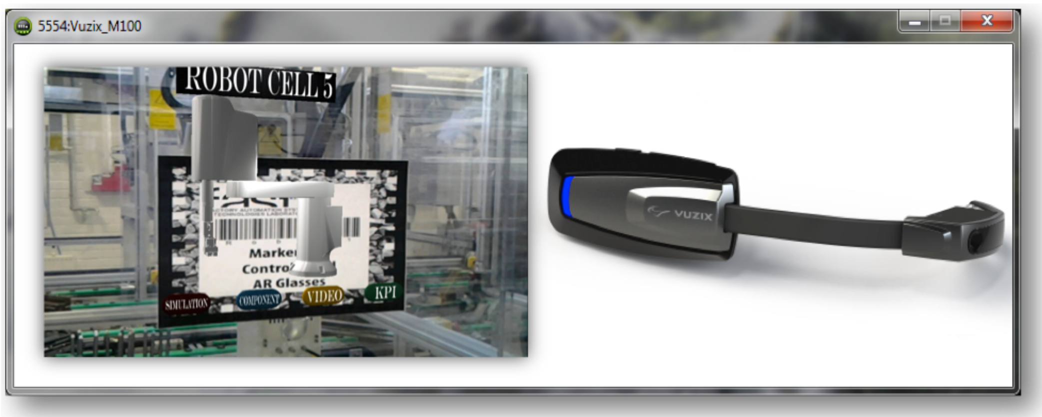


Figure 4.22: Smart Glasses Emulation Device with Hardware Vuzix Smart Glasses.

Finally, the wearable interface has the same functionalities of AVITA described in previous versions. In addition, the virtual buttons are added in order to create a different interaction with the user, in this example, the user can select the same elements based on the position of the user finger and the obstruction of the virtual object over the AR pattern. Figure 4.23 shows the same transition after the selection using the virtual interface.

Figure 4.23: Smart Glasses Interfaces based on Vuzix Smart Glasses Hardware.

4.4.2 Version Control

The initial stage test of AVITA does not includes the scenario for MS, the principal objective designed in order to prove of the concept integration with mobile devices. So far, different solutions are found to implement the application with many technologies described with more details in Chapter 2. After several tests implementing and analyzing many technologies a solution was found, taking the requirements for the scenario description. The selection includes the combination with UNITY3D and Vuforia AR. Table 4.2 shows the results of the first test of the MAR application.

Table 4.2: Initial Test AR application

Achievements	Description	Results
Tech. Selection	Comparison between different frameworks using AR engine.	UNITY3D & Vuforia AR
Tech. Integration	Install compatible software in order to integrate the technologies selected.	Creation of simple 3D content.
Tracking Algorithm	Selection of the Tracking algorithm and AR marker.	TrackingAR Library using default markers
AR marker	Personalize AR marker based on examples.	Using design tools the AR was created.

After the initial test, more elements were included in order to achieve the objectives of the scenario description and ASTUTE project. However, not all the results were positives, some drawbacks are found in the integration of the basic elements others are found in other stages when the application was becoming more complex.

The second test includes more elements and integration of more technologies, some inside the architecture of the MAR others as external sources creating and exporting elements to the UNITY3D engine. Table 4.3 shows the new achievements and improvements of the application, including the results of these tasks.

Table 4.3: Additional Requirements Test AR application

Achievements	Description	Results
Augmented Robot	Importing 3D content to the AR engine.	The 3D components are assembled.
Scripting in UNITY3D	Adding programmable actions to the 3D content.	A loop containing the Robot's kinematics.
Augmented Information	Integration of 2D & 3D content in main screen.	Basic demo integrating 3D content.
3D Environment Design	Personalize 3D content in UNITY3D.	Enviroment quality added lights and textures.

The drawback in the second test was the lack of design rendered in Blender. The user does not have elements to interact such as the interface. For this reason, the application includes a programmable loop creation the simulation of these actions. So far, the interactions are incorporated in this version for testing purposes such as animations, text and components.

Finally, before the general usability test the second version is improved, integrating more elements and functionalities into the AR application. In fact, one of the main elements integrated was full interaction with the user after fixing some problems with UNITY3D engine and the interaction between scenes.

As a result, the solution proposed was the creation of the application in only one scene creating the programmable interaction with all the elements at the same time turning off and on the elements displayed in order to avoid the delays created in previous versions in the application using many scenes. Table 4.4 shows the results before the usability test.

Table 4.4: Final Requirements Before Usability Test

Achievements	Description	Results
Augmented Robot	Importing new 3D content into the AR engine.	3D environment completed based on specification.
GUI elements	Generation of user interface based on the new elements.	User Interface created.
Video	Display relevant information of the status of the system.	The 3D element containing video of current status.
KPIs	3D & 2D content generated based on actual status.	Graphs are integrated, dashboard of current status.
Manuals	Generation of manuals based on the current status.	Display PDF sheets, equipment sheets.
User experience	Programmable content animations creating friendly interaction with the user.	The script contains intersection to modify 3D content in the scene.

In this final stage, the application starts in a programmable instruction, waiting for the user response based on the GUI created specially for the elements integrated in AVITA. The application contains a dashboard of the status of the robot, additional to elements such as video, KPIs and manuals. At the end of the test process different problems discussed in previous versions are solved including the interaction with the user, the lack of 3D environment design and the delays using the application in run-time.

5. RESULTS

In Chapter 3 is described the sequential process used into the development of a MAR application. Also, within these tasks is included the analysis and the design of the application in order to achieve better results. After many interactions with the sequential development process (Spiral development) some values can be retrieved from the implementation part, feedback given in meetings related with project ASTUTE and the usability test. Summing up, this Chapter will describe general and particular information about the results of each testing stage during the sequential process, until the creation of a stable version of the MAR application called AVITA.

As is mentioned in previous Chapter 3 and 4 the application is created based on basic concepts of AR after the design and analysis new requirements were integrated into the application, including the version control of AVITA and the different schemes of development. Meanwhile, the scenario was defined with few functional requirements in order to active real objectives at the beginning of the project. On the other hand, after the implementation and interaction with several processes in the sequential method, the version of AVITA was improved, due the fast feedback and the integration of the SysML diagrams. This Section describes the results of the test phase during the integration of the sequential methodology in particular the results displayed by these tests.

5.1 Test Results

5.1.1 Usability Test

The Usability Test was implemented in order to evaluate the product created along the process of development of ASTUTE project. The version presented in this usability test integrates the entire ASTUTE solution for the manufacturing domain which is described in detail inside the deliverables of the project. The test was divided into different sections one of those including the Mobile Augmented Reality (MAR) application in different stages. The test starts from the internal feedback with staff of the FAST. After some improvement, the version was ready to be applied in the real usability test. Figure 5.1 shows the user in the usability test using the MAR application.

Figure 5.1: AVITA Usability Test: Image taken from the real test at FAST.

The information retrieved by the usability test can be used to improve new versions of AVITA. During the session test, the application with augmented reality was accepted by the user, receiving positive feedback about the application. The data in general was divided into different sections. The information included in this document only takes the information related the usability test with the MAR application. Figure 5.2 shows AVITAs main screen used in the usability test process.

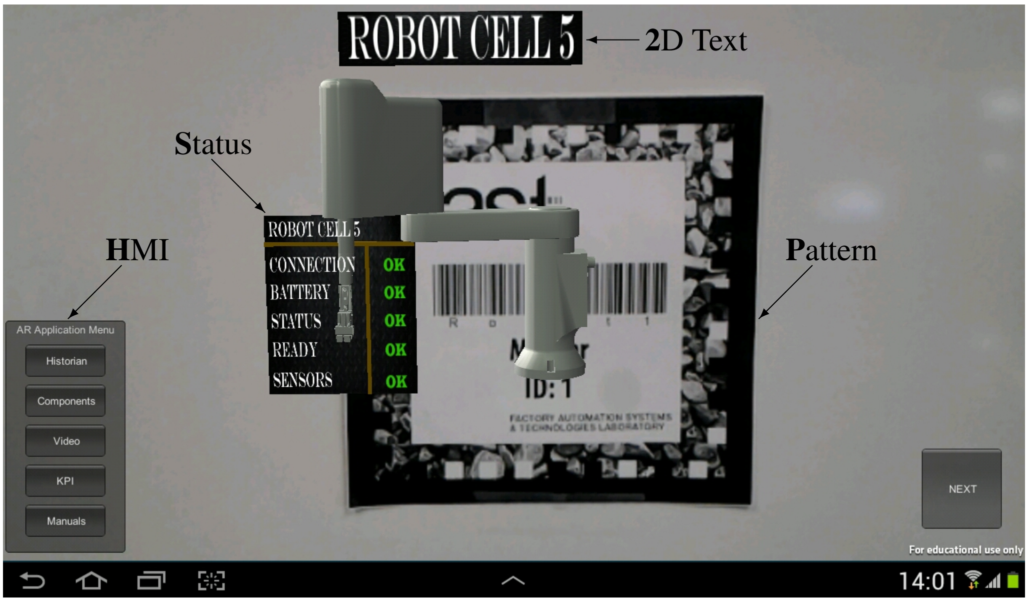


Figure 5.2: AVITA Interface for Usability test.

5.1.2 Monitoring Usability Test

The usability test was integrated by an experimental design. The methodology of formative testing is described in the following list.

- Thinking aloud method: participants think aloud while performing the tasks. The post-test questionnaire consists of:
  - Open questions
  - Standard SUS questionnaire
  - Satisfaction questionnaire

This methodology was used for the entire application for ASTUTE project. Also only the relevant information in the section of AR is taken into account to improve the application. The following format is described based on the real usability test task named "Augmented Reality".

Please point the device to the QR code. You can move the device to change the angle of the view.

1. Explain me what you see.
2. What information do you expect behind the following menus (do not touch yet):
   - Historian
   - Components
   - Video
   - KPI
   - Manuals
   - Next
3. You would like to see how the different parts of the robot look like, please look at them.
4. What is the value of Overall Equipment Effectiveness (OEE) in the Key Performance Indicator (KPI)?.
5. Please press next. Which chart do you prefer? Why? Test between 2D and 3D charts

6. You would like to see how the robot does its normal cycle, please watch a video about it. If the video isn't short it can be stopped after a moment so this task won't take too long.

7. Name three things you would change in this view.

5.2 System Evaluation

This document will describe the evaluation after the usability test including positive and negative feedback about the implementation of AVITA using the MS. As is mentioned in this Chapter only some recommendations are found during the test.

Summary statistics

As a result, Table 5.1 shows the compilation of information taken the results from the usability test.

Table 5.1: Usability Test AR Summary Statistics

Task	**T3**	**T4**	**T6**
Completion	8	11	11
Assists	1	1	0
Errors	3	0	0
Average time on task	82 sec	56 sec	22 sec

The following list contains the observation during the test with AVITA.

- Few of the participants did not know what historian means. Otherwise the terms in the menu were understood correctly.
- Many of the participants did not realize to move closer or change the angle of the device to see the things in the screen better.
- The errors came from the participants not realizing to push the next button. However, note that is possible that the users did not see the button. In fact, some of them mentioned that the next button should be positioned near the other buttons.
- It should be noted here that the scenarios chosen for the AR part were not representing AR it in the best possible way. These scenarios should be rethought for the next test.

Finally, Table 5.2 shows the most common answers to the question "Name three thing you would change" in the section of AR.

Table 5.2: Most common user's answers in usability test

Answer
Improved graphics (2 participants)
Zoom functionality (3 participants)
Keep the information on the screen when pointing away from the QR code (3 participants)
Changing the background colour (2 participants)
Graphs not fit for 3D environment (1 participants)
Next button not easy to find (1 participants)

Design recommendations

The subsection was created in order to organize the priority in the design process. In other words, based on the feedback of the users the information is collected and analyzed. Taking the data from AVITA usability test, the following list was created describing the design recommendations based on the results of the usability test.

- AR QR code should be covered with one colour.
- AR buttons should be grouped together (next-button should be next to the other buttons).
- It should be considered which elements need to be shown in 3D format and others should be shown in a normal screen.
- Keep information on screen when pointing away from QR code.
- AR zoom functionality.

6. CONCLUSIONS

The MAR application is applied successfully in manufacturing domains. The creation of AVITA shows that it is possible the creation of applications based on AR for different domains. In the process of development and design there were some issues with integration and debugging. Problems that were fixed and improved in the last version of AVITA, due the interactions with the layers of the software development process.

The MAR application is designed for monitoring manufacturing systems in specific user scenarios in the factory floor. The basic implementation of AVITA is designed based on marker-based AR applications including: the multi-target tracking, the AR Interface, the real time information and the delay between the user actions. In addition, the initial version it does not include modules of user experience and user interactions, but was designed for integrating several modules and scenes for future developments.

The final stage of AVITA contains and solves different issues found along the development process, also new requirements are added in the process creating more complex scenarios in the application. AVITA is capable of displaying 3D content for the different type of users involved in the factory floor showing specific information based on the user action or tasks that the system requires. The patterns were placed around the factory floor and are integrated into the MAR application detecting different types of markers making possible the display of different element along the different tasks in the factory floor. The user is able to see the virtual content within different elements, for example, Videos, Manuals, KPI and Components.

The 3D content is animated based on the user interaction with the application. The user interface is simple, functional and easy to use. In comparison with the mentioned interfaces created in this document. The size of the interface changes according to the selected device. In addition, the version for wearable devices includes other solution in the pattern to select the action with the user finger creating a real interaction with the real world between the user and the MAR application. Finally, the integration with other projects and technologies makes AVITA a great application to be improved and adapted for more complex scenarios with different Manufacturing Systems.

6.1 Further Work

Along the different interactions with the layer of the implemented methodology for software development (see Chapter 3), more requirements and future tasks were found. After the development of the interface for wearable devices (Vuzix Smart Glasses M100) new functionalities can be added for future development of AVITA. The following list describes in detail the further improvements or integrations with other technologies.

1. Integrate new functionalities based on the usability test results (see Chapter 5).

    - Graphic improvements: Include the details of the virtual content displayed in the user interface.
    - Zoom selection: Integrate a module to increase and decrease the size of the virtual content, if the user selects this option. In particular, for mobile devices this feature will be added. On the other hand, for smart glasses the zoom is added based on the position of the user, feature already added in the last version of AVITA.
    - Keep the information on the screen even if the AR is out of focus.
    - Changing the background color: Due the MAR application is marker-based AR application, the versions using the AR marker will stay at it is. The new feature is the integration of marker-less AR application in order to achieve the problem with the size of the pattern.
    - Graphs not fit for 3D environment: Some of the elements shown in the application are in the design and testing phases. Then, the information displayed will be integrating 2D images and the 3D elements will be improved for future versions. As a result, the user can selects between 3D and 2D content.
    - Next element is not easy to find: The user interface will be improved for some versions of AVITA.

2. Integration of new functionalities based on implementation process and development point of view.

    - Mobile information: Add functionalities of the mobile device to acquire position and acceleration in order to interact with the virtual content, inside and outside of the factory floor.
    - Create and design more complex scenarios for the users involved in the factory floor.
    - Integration of more complex manufacturing systems in order to integrate real information from production line.

- Based on the version created for Vuzix smart glasses, integrate voice recognition and mobile sensor fusion.
- Prepare AVITA for new devices and different OS. For example, iOS devices.

Finally, the testing and debugging process will be repeated after find more requirements and adapt different models of AVITA. Thus, more complex usability test will be integrate new versions of the MAR, also the entire developed for ASTUTE project. The application will be presented in future meeting to propose general solutions for monitoring manufacturing systems using AR.

BIBLIOGRAPHY

[All08] Open Handset Alliance. Android open source project: Android sdk. http://developer.android.com/sdk/index.html, 2008.

[All09] Open Handset Alliance. Android open source project: Android ndk. http://developer.android.com/tools/sdk/ndk/index.html, 2009.

[ART04] ARToolworks. Artoolworks computer tacking library. http://www.artoolworks.com/, 2004.

[BC03] Grigore C. Burdea and Philippe Coiffet. *Virtual Reality Technology*. John Wiley and Sons, 2003.

[BKLP04] Doug A. Bowman, Ernst Kruijff, Joseph J. Laviola, and Ivan Poupyrev. *3D User Interfaces: "Theory and Practice"*. Pearson Education, 2004.

[Bri96] John C. Briggs. The promise of virtual reality. *The Futurist (Magazine/Journal)*, 30, 1996.

[Bri12] BrilliantServiceCorporaction. Smartphones are outdated. http://www.brillianservice.co.jp/viking/, 2012.

[Bry] Steve Bryson. Ames research center. http://www.nas.nasa.gov/Software/VWT/vr.html.

[CM92] T.P. Caudell and D.W. Mizell. Augmented reality: an application of heads-up display technology to manual manufacturing processes. In *System Sciences, 1992. Proceedings of the Twenty-Fifth Hawaii International Conference on*, volume ii, pages 659–669 vol.2, 1992.

[Com99] Visual Components. 3d factory visualization tool. http://www.visualcomponents.com/, 1999.

[Cor12a] Vuzix Corporation. Smart devices fom ces 2012. http://www.vuzix.com/news/productphotos.html, 2012.

[Cor12b] Vuzix Corporation. Vuzix m2000ar:first ar system for end-user use in industrial enviroments. http://www.vuzix.com/markets/blog.html, 2012.

[Cor13] Oracle Corporation. Java se development kit. http://www.oracle.com/technetwork/java/javase/downloads/index.html, 2013.

[FP11] Valentino Frati and Domenico Prattichizzo. Using kinect for hand tracking and rendering in wearable haptics. *World Haptics Conference (WHC)*, 2011.

[Fun12] Eclipse Fundation. Eclipse: Integration with indigo version. http://www.eclipse.org/, 2012.

[Fur13] Rod Furlan. Build your own google glass: A wearable computer that displays information and record video. *IEEE Spectrum*, page 2, 2013.

[FYJ10] Joshua Fabian, Tyler Young, and James C. Peyton Jones. Integrating the microsoft kinect with simulink: Real-time object tracking example. *IEEE/ASME TRANSACTIONS ON MECHATRONICS*, 2010.

[Gar12] Gartner. Gartner's 2012 hype cycle for emerging technologies identifies "tipping point" technologies that will unlock long-awaited technology scenarios. http://www.gartner.com/newsroom/id/2124315, 2012.

[Gog12] Goggle. Project glass. https://plus.google.com/+projectglass/posts, 2012.

[IBM81] IBM. Ibm rational software. 2003, 1981.

[Inc11] Invensense Inc. Invensense: Monition processing library. http://www.scantec.de/en/knowledge-base/technology-transfer/2011-1/mpu-6050-gyroskop-and-accelerometer-in-a-single-chip.html, 2011.

[Int12] Visual Paradigm International. Visual paradigm for uml. http://www.visual-paradigm.com/, 2012.

[Jun09] Junaio. Augmented reality bowser desegned for 3g and 4g mobile devices. http://www.junaio.com/, 2009.

[Kat99] Hirokazu Kato. Artoolkit library for building augmented reality applications. http://www.hitl.washington.edu/artoolkit/, 1999.

[KR13] Gregory Kipper and Joseph Rampolla. *Augmented Reality: An Emerging Technologies Guide to AR*. Syngress Publishing, 2013.

[Lew09] William E. Lewis. *Software Testing and Continuous Quality Improvement, Third Edition*. Auerbach Publications, December 2009.

[Lib09] Sologico Libre. Aithoring tool: Augmented reality and mapping. http://www.sologicolibre.org/projects/atomic/en/, 2009.

[Mad11] Lester Madden. *Professional Augmented reality browser for smartphones: Programming for junaio, Layar and Wikitude*. Wrox, 2011.

[Met03] Metaio. Metaio:the augmented reality company. http://www.metaio.com/, 2003.

[Mot12] LEAP Motion. Leap motion controller. https://www.leapmotion.com/, 2012.

[Mul11] Tony Mullen. *Prototyping Augmented Reality*. Sybex, octuber 2011.

[NDI+11] Richard A. Newcombe, Andrew J. Davison, S. Izadi, P. Kohli, Otmar Hilliges, J. Shotton, David Molyneaux, Steve Hodges, David Kim, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. *Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on*, pages 127–136, Oct 2011.

[NLLC13] Oscar Neira, Angelica Nieto Lee, Jose. L. Martinez Lastra, and Roberto S. Camp. A builder for adaptable human machine interfaces for mobile devices. *INDI 2013*, 2013.

[oF12] VTT Technical Research Center of Finland. Alvar - a library for virtual and augmented reality. 2012, 2012.

[PÍ2] Héctor Garcá Pájaro. A 3d real-time monitoring system for a production line. Master's thesis, Tampere Universisty of Technology, May 2012.

[QDL13] QDLaser. Laser head set. http://www.qdlaser.com/, February Pending[Only exibitions], 2013.

[Qua12] Qualcomm. Qualcomm technologies: Vuforia augmented reality. https://www.vuforia.com/, 2012.

[SOG03] R. Silva, J. C. Oliveira, and G. A. Giraldi. Introduction to augmented reality. Technical report, National Laboratory for Scientific Computation, Brazil, 2003.

[Sol95] Cygnus Solutions. Unix enviroment command line interface for woindows. http://cygwin.com/, 1995.

[SVH+02] Tim Suthau, Marcus Vetter, Peter Hassenpflug, Hans peter Meinzer, and Olaf Hellwich. A concept work for augmented reality visualization based on a. In *Medical Application in Liver Surgery, Technical University Berlin, Commission V, WG V/3*, 2002.

[Tec05] Unity Technologies. Unity 3d: Cross-platform game engine with a built-in ide. http://unity3d.com/, 2005.

[Tec12] Unity Technologies. Unity 3d: Documentation manual, scripting reference. http://docs.unity3d.com/Documentation/ScriptReference/Transform.Rotate.html, 2012.

[TS13] Scoot Torborg and Start Simpson. What's inside google glass. http://www.catwig.com/google-glass-teardown/, Jun 2013.

[vdKBLF09] Raimo van der Klein, Claire Boonstra, and Maarten Lens-FitzGerald. Layar browser augmented reality. http://www.layar.com/, 2009.

[Vuz12] Vuzix. Smart glasses m100. http://www.vuzix.com/consumer/productsm100.html, 2012.

[Wag07] Daniel Wagner. Handled augmented reality. Dissertion, Graz, Astria, Octuber 2007.

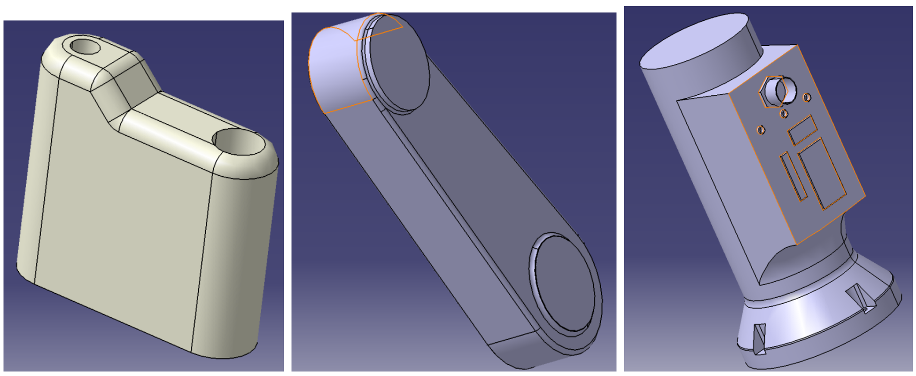
[Wik08] Wikitude. Wikitude world largest augmented reality sdk. http://www.wikitude.com/, 2008.

A. APPENDIX

A.1 CAD/CAM Integration Module

The AR application combines the displayed signals from the real world and created objects in 3D. For this reason, to create applications with AR it is needed the generation of environments in 3D. Currently, there are different solutions to create 3D object. In this example, some CAD tools were used to create and export the 3D environment to the Cross-Platform Game Engine (UNITY3D). The 3D objects can be created in different types of 3D graphic softwares. For example, DELMIA, Blender, Maya 3D and Solid Works. In particular, for the application design of AVITA the 3D objects are related with a manufacturing floor and based on that specially scenario described in Chapter 3.

The created AR elements are designed using CATIA and DELMIA generating the complete robot and components such as the base, the platform of the robot arm and the gripper. The objects were designed and created based on the real robot installation of FASTory shows in Figure 3.2. In addition, the Figure A.1 shows the elements created in DELMIA 3D software.



(a) (b) (c)

Figure A.1: Virtual environment development, generation of 3D elements. (a) Platform with sensors and motors. (b) Platform Joint. (c) Robot Base and connectors.

After the creation of the robot components, the integration can be created on one or several pieces. The Figure A.2 shows the integration of the 3D objects and the model. Thus, the complete model is ready to be exported in UNITY3D.

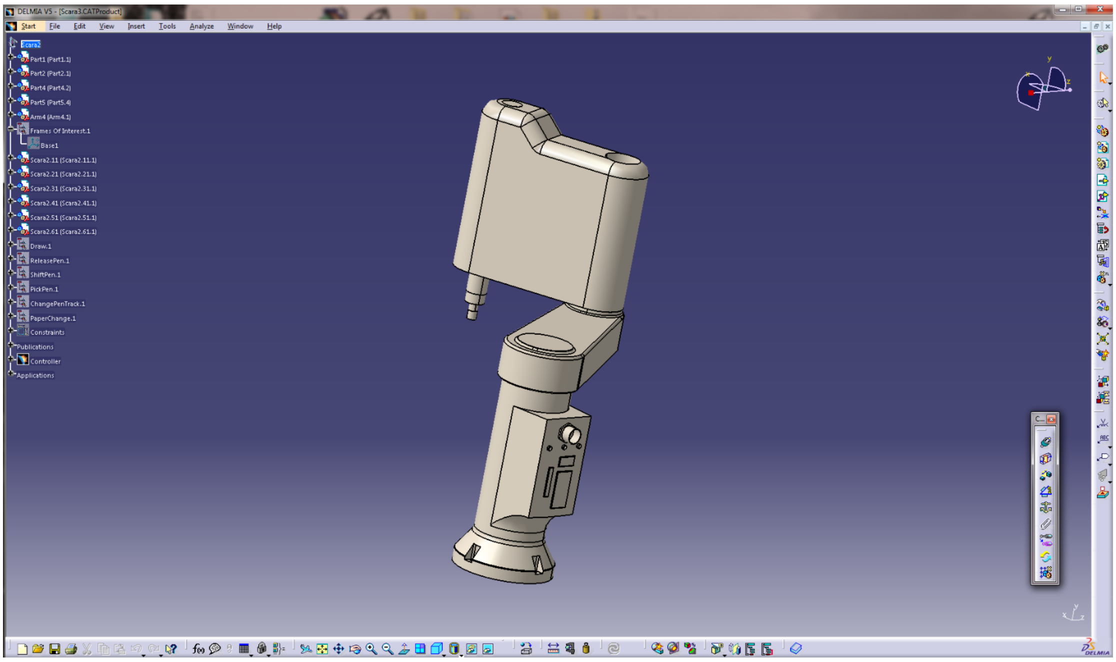


Figure A.2: 3D object assembled completed in DELMIA 3D CAD software [PÍ12].

Finally, the environment is complete and could be exported. Moreover, UNITY3D has an option to import objects or elements from different sources. For instance, in the implementation of AVITA, DELMIA and Blender are used to export the 3D environment in a compatible format for UNITY3D. Thus, UNITY3D is able to integrate them and Figure A.3 shows the basic integration of the elements in UNITY3D.

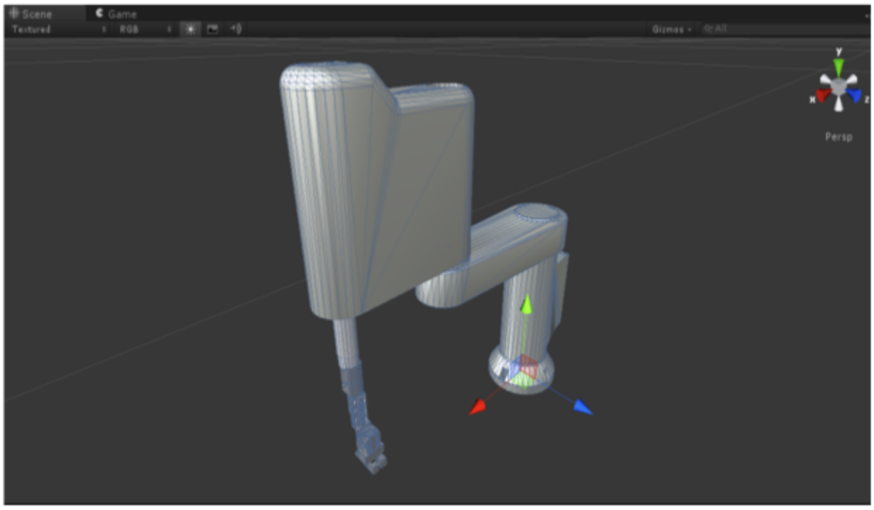


Figure A.3: UNITY3D assembling 3D environment.

B. APPENDIX

B.1 Cross-Platform Game Engine (UNITY3D) Integration with AR.

The information presented in this Appendix covers the integration with UNITY3D & Vuforia libraries. However, the entire setup is followed by five steps and some of them are shown in the developer web page of Qualcomm Inc. others are implemented implemented only in this thesis work.

Step 1: Installation.

The initial step is download the package for your development platform, in the case of AVITA import the compatible files to UNITY3D. Then, after the download, open a new project in UNITY3D importing the files vuforia-android package available in the main page of Vuforia [Qua12].

Step 2: Compile a simple project.

After installing the development package and imported the files into UNITY3D.

Then, create a project avoiding spaces in the project name. Next, add a data sheet to the project.

1. Creating a target on the Target Management System (TMS).
2. Using existing targets from other projects.

After the package with the textures is assigned. Then, the Targets are created inside the UNITY3D engine.

Also, It is needed copy a replace the files as is shown in Figure B.1.

Until now, the Vuforia AR extension for UNITY3D is imported and ready to create a new project in AR.

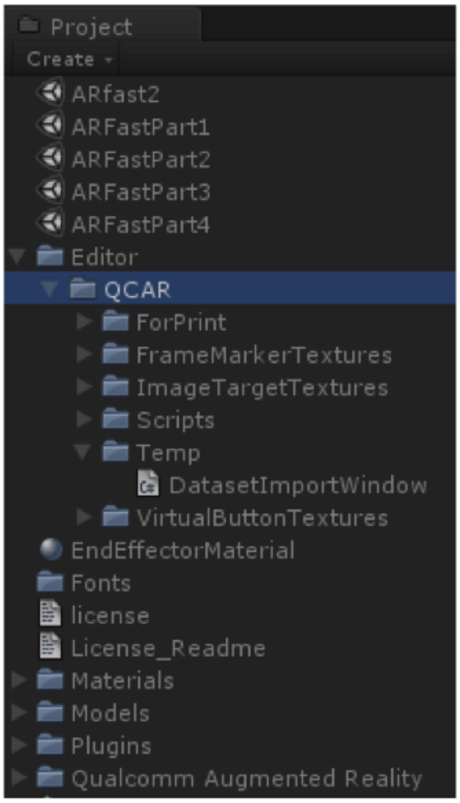


Figure B.1: Importing textures to the existent project.

Step 3: Create an AR camera inside the project.

- Expand the Qualcomm Augmented Reality (AR) folder, and then expand the Prefabs folder.
- Delete the Main Camera. Then, drag an instance of the ARCamera prefab into the scene. In the end, The ARCamera is responsible for rendering the camera image in the background and can manipulate the scene with 3D objects reacting to the tracking data.

With the ARCamera in place and the target assets available in the StreamAssets/QCAR folder, is possible the deployment of the application. The Figure B.2 shows the final configuration of the camera inside UNITY3D.

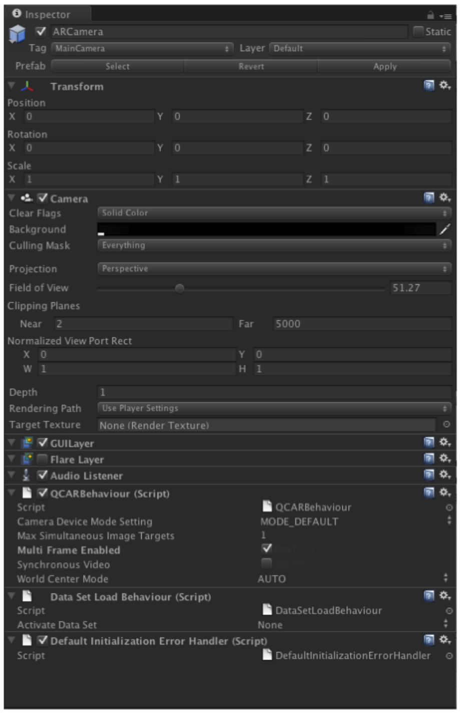


Figure B.2: ARCamera's Inspector view.

Step 4: Trackable Image.

After the set up of the camera is ready, is needed the insertion of the image target prefab into the scene in UNITY3D and is decried in the following list:

- Select the Image Target object in the scene. In particular, the with the guide given in Chapter 3 in the Figure 3.12.
- Select the inspector view and check the option of the Trackable behavior attached in the project, with a property named Image Target (this property contains a drop-down list of the entire available Image target).
- Then, select the Data Set and Image Target from the StreamingAssets/QCAR folder.

In this phase it is possible the integration of your own pattern as is shown in Figure B.3. In addition, the main page of the of Vuforia ARlibraries [Qua12] explains in detail the creation a better trackable images, also included in Chapter 3.

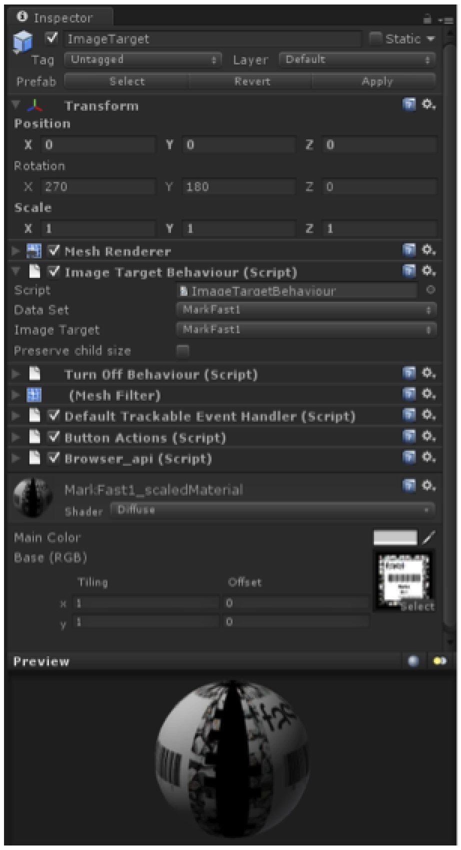


Figure B.3: Image Target Selection.

Step 5: Testing & Animation Mode.

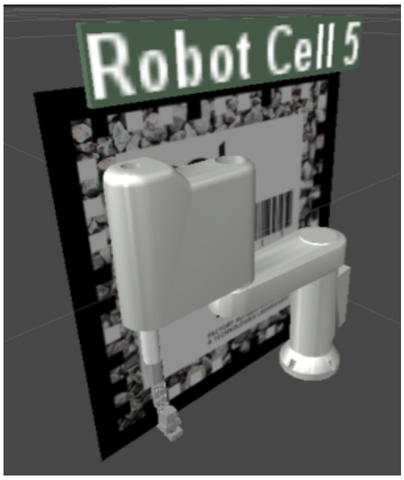
Finally, after the installation of the development AR package, the 3D environment is imported into UNITY3D engine and the trackable image also is in position according to the 3D environment and the AR-Camera. Thus, is possible make some test based on the results in the animation mode provided by UNITY3D.

The animation mode is a preview of the application used for testing purposes and fast debugging. The animation mode help to see if any of the steps mentioned before was missing or if the interaction with the animation is incorrect. The correct way to use this functionality is to check malfunction in the code or in any of the stages of development.

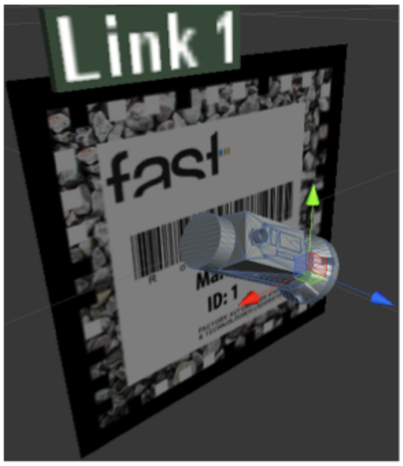
Figure B.4(a) shows how the 3D object (robot and text) and the pattern image are integrated in UNITY3D. This version was proving the concept of AR with extension in UNITY3D at the same time integrating elements for industrial environments.

Then, the animation mode is able to simulate the user's actions working along the entire application. For example, the GUI created in this mode can be tested, in order to see the responses with the user's actions and the response in the scene.

Figure B.4(b) shows the example in scene mode, that can be modified in the graphical interface to change the object properties. On the other hand, Figure B.4(c) shows the example in animation mode for debugging purposes.



(a) 3D object & Trackable Image in Animation Mode.



(b) UNITY3D Scene Mode.

(c) UNITY3D Animation Mode.

Figure B.4: Initial Implementation test. (a) 3D object & Trackable Image. (b) UNITY3D Scene Mode. (c) UNITY3D Animation Mode.

C. APPENDIX

C.1 Robot Kinematics, User interface & Animation Interaction

The interaction with the user was one of the main objectives of the entire implementation of AVITA, The user have different options that can take in order to see more information related with the use case as was mentioned along this thesis work. Moreover, one of the initial test for AVITA was the integration the kinematics of the robot. Then, the interaction with the user was added using a general GUI finally solving some problem with the scene flow inside the UNITY3D the generation of a sequence of animation based on the actions sent by the used was made, in order to improve the interaction with the user and the application.

Robot Kinematics and Code Generation

To generate the rotation of each piece of the Robot some functions are called, in order to start the script with the functions when the application is in Animation mode or installed in the device. Moreover, The scripts also contains another function that is called when the animation change (by frame). The function **update** that it is executed as a buckle considering the animation frames. In other words, the instructions inside the function are executed each frame. In particular, Figure C.1 describes the instruction transform.rotate (function Rotate (xAngle : float, yAngle : float, zAngle : float, relativeTo : Space = Space.Self) : void) [Tec12] applying a rotation of eulier Angles.z degrees around the z axis, following the order around the z, x and y axis, containing elements that create the effect of rotation of the object.

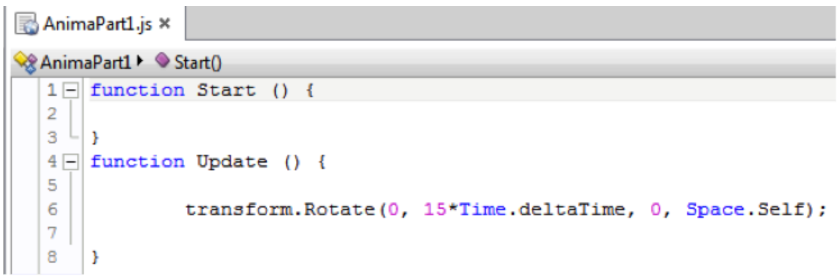
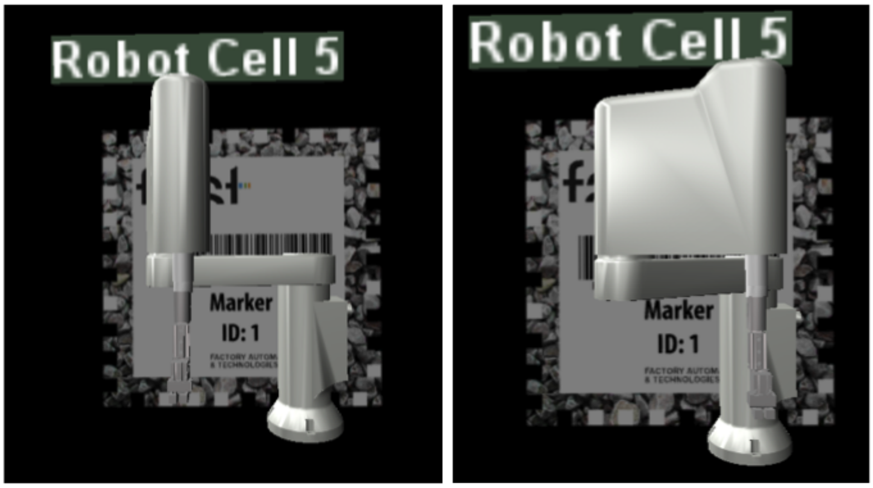


Figure C.1: Function to Transform and Rotate in UNITY3D.

Finally, to see a complete example of the robot's kinematics applying the function defining in UNITY3D. The Figure C.2(a) shows the Robot initial position before the function **update** starts calling the methods implemented above. Figure C.2(b) shows the Robot after the frame change the position of the robot. In fact, this code is triggered after the user select one of the option in the interface.



(a) UNITY3D Animation Mode Initial Position. (b) UNITY3D Animation Mode Transform Function.

Figure C.2: Images Taken from the initial test of the implementation of the kinematics of robot. (a) UNITY3D Animation Mode Initial Position. (b) UNITY3D Animation Mode using the functions to Transform and Rotate elements.

As a result, the rotation and translation are created for each scene. Then a relationship between the scripts and the object can be made in a graphical option in UNITY3D. In addition the following list define the main functions of the methods called in UNITY3D.

- **Update**: this function is called before rendering a frame. This is where most game behavior code goes, except physics code [Tec12].
- **FixedUpdate**: this function is called once every physics time step. This is the place to do physics-based game behavior [Tec12].
- **Code outside any function**: code outside any functions is run when the object is loaded. This can be used to initialize the state of the script [Tec12].

The following list describes with more details the creation of methods and functions inside UNITY3D. For the example, in the implementation of the robot's kinematics, JavaScript was used to create the animation of the components. On another hand, to create the general GUI and the animations for the interaction of the user and the application, C# was integrated. Another capability of UNITY3D is that is multilanguage platform.

1. Create a connection with the Game Object and the script: The script should be saved it as .js format. Then, drag and drop the script inside the object, after that the script created can be used for that scene applied to that specific object. Figure C.3 shoes the relationship between the Game Object created in unity and the scripts.

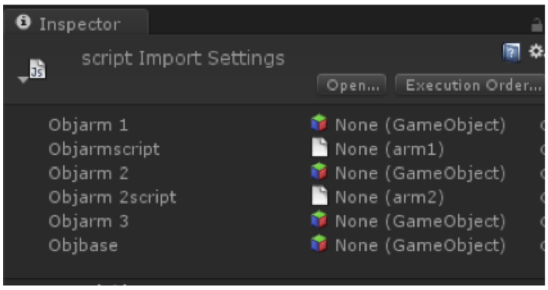


Figure C.3: Relationship between Game Objects & Scripts.

2. Open the document inside UNITY3D : the scripts are saved inside the UNITY3D engine, the the scripts can be modified inside the same project. In other words, UNITY3D works as a Integrated Development Environment (IDE) (MonoDevelop-UNITY), making possible the modification of the script in the same environment.

3. Script Structure

   - Declaration:The structure inside a script inside UNITY3D is similar to one in JavaScript. Starting from the variables declaration and ending from the Main function. The Figure C.4 shows the declaration of object and variables.

```
#pragma strict

import System;

var objarm1: GameObject;
var objarmscript: arm1;
var objarm2: GameObject;
var objarm2script: arm2;
var objarm3: GameObject;
var objbase: GameObject;

var state= new boolean[6];
var statetrig= new boolean[6]

var x0:float;
var y0:float;
var x1:float;
var y1:float;
var speed:float=0.2;

var a:float;
var baux:float;
var b:float;
var slider1: float = 0.0;
var slider2: float = 0.0;
var sliderx: float = 0.0;
var slidery: float = 0.0;
var c:float;

var currentAbsTime:DateTime;
var counterTime:TimeSpan;
var zeroTime:DateTime;
var j:int=0;
var counterTimeSec:float;
```

Figure C.4: Import & declaration instructions.

- Start: Figure C.5 shows the generation of the Start method in UNITY3D.

```
function Start () {

    //find other robot GameObjects
    objarm1=GameObject.Find("rob_arm1");
    objarmscript=objarm1.GetComponent(arm1);
    objarm2=GameObject.Find("rob_arm2");
    objarm2script=objarm2.GetComponent(arm2);
    objarm3=GameObject.Find("rob_arm3");
    objbase=GameObject.Find("rob_base");
    state[0]=true;
}
```

Figure C.5: Main Function in UNITY3D.

- Update: Figure C.6 shows the method called each frame checking the actual position of the robot, based on that the method used move the components creating the robot's kinematics

```
currentAbsTime=System.DateTime.Now;
counterTime= currentAbsTime - zeroTime;
counterTimeSec=counterTime.TotalSeconds;
if ((Mathf.Abs(sliderx - x1)>= 0.01) || (Mathf.Abs(slidery -y1)>= 0.01)){
    sliderx=x0 + speed*counterTimeSec*(x1-x0);
    slidery=y0 + speed*counterTimeSec*(y1-y0);
}
slider2= (Mathf.Acos((Mathf.Pow(sliderx, 2.0) + Mathf.Pow(slidery, 2.0) - Mathf.Pow(0.348, 2.0) ·
a=(Mathf.Atan2(slidery, sliderx))* Mathf.Rad2Deg;
baux= (0.25*Mathf.Sin(slider2* Mathf.Deg2Rad))/(0.348 + 0.25*Mathf.Cos(slider2* Mathf.Deg2Rad));
b= (Mathf.Atan(baux)) * Mathf.Rad2Deg;
slider1= a-b;

objarm1.transform.localEulerAngles.y= - slider1;
objarm2.transform.localEulerAngles.y= - (slider2 + slider1);
objarm1.transform.localEulerAngles.x= 0;
objarm2.transform.localEulerAngles.x= 0;
objarm1.transform.localEulerAngles.z= 0;
objarm2.transform.localEulerAngles.z= 0;
```

Figure C.6: Update Function UNITY3D.

D. APPENDIX

D.1 Deployment in Mobile Devices & Smart Devices

The last sequential step is the deployment of an application, after testing and debugging the application created in UNITY3D using the Animation mode some general steps are needed in order to deploy and compile the application. In general, if the appendices described in this document A, B, C, D are followed step by step also the animation looks like the previous Figures. Then, it is possible compile the AR application in a mobile device or wearable devices. The Figure D.1 shows different Build options in UNITY3D. The first step is the selection of Android. Then, select the option Player settings.

Figure D.1: Build Options in UNITY3D.

Now, before build and run the application it is needed the selection of Player-Setting and change some of the mandatory values. The list shows the general information that is required to compile the application in the device.

- Identification.
  - Bundle Identifier: generate the name of the library for the android application.
- Bundle Version: the version of the application.
  - Configuration.
  - Graphic level: by default using OpenGL ES 2.0 but this can be changed it.

Then, the other options such as: Resolution & Presentation, Icon Splash Image and Publishing Settings can be stay empty and get default values.Thus, Figure D.2 shows the general information of the Player setting.

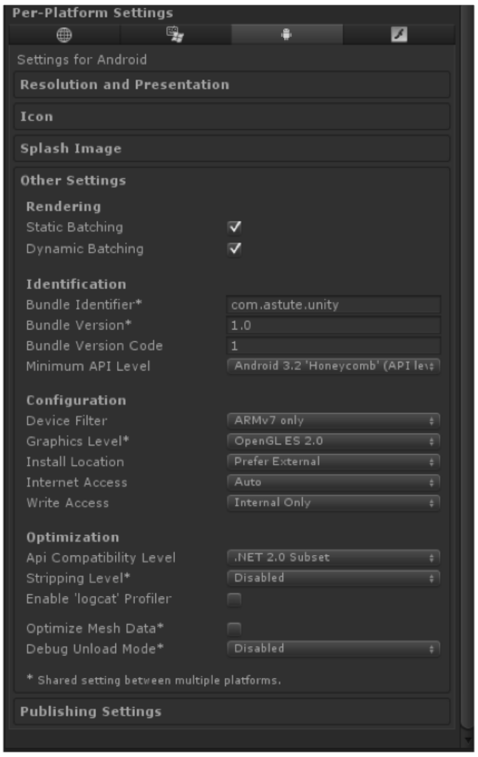


Figure D.2: Player Settings in UNITY3D.

Finally, connect the device to the PC and wait until the device is installed and ready to be used. Now, the project is ready to be compiled and deployed. As is mentioned in previous steps the option of Build & Setting is selected, instead of Player setting. Figure D.3 shows the last action to create and AR application integration Vuforia libraries and UNITY3D

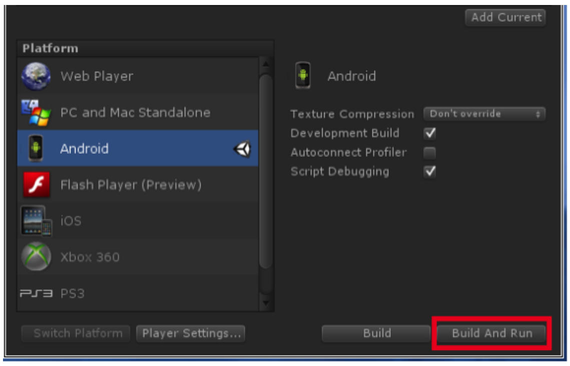


Figure D.3: Build & Run Configuration in UNITY3D.

E. APPENDIX

E.1 Sequential Diagrams (User-Application Interface Interactions)

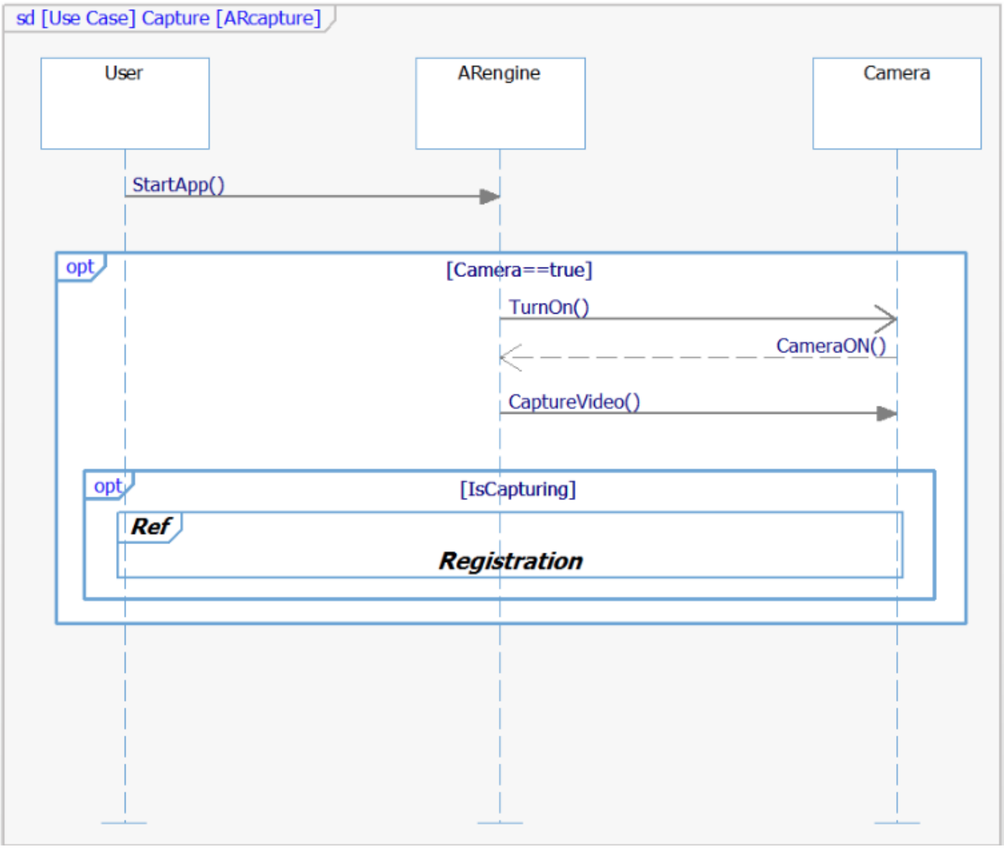


Figure E.1: AR Capture Sequential Diagram

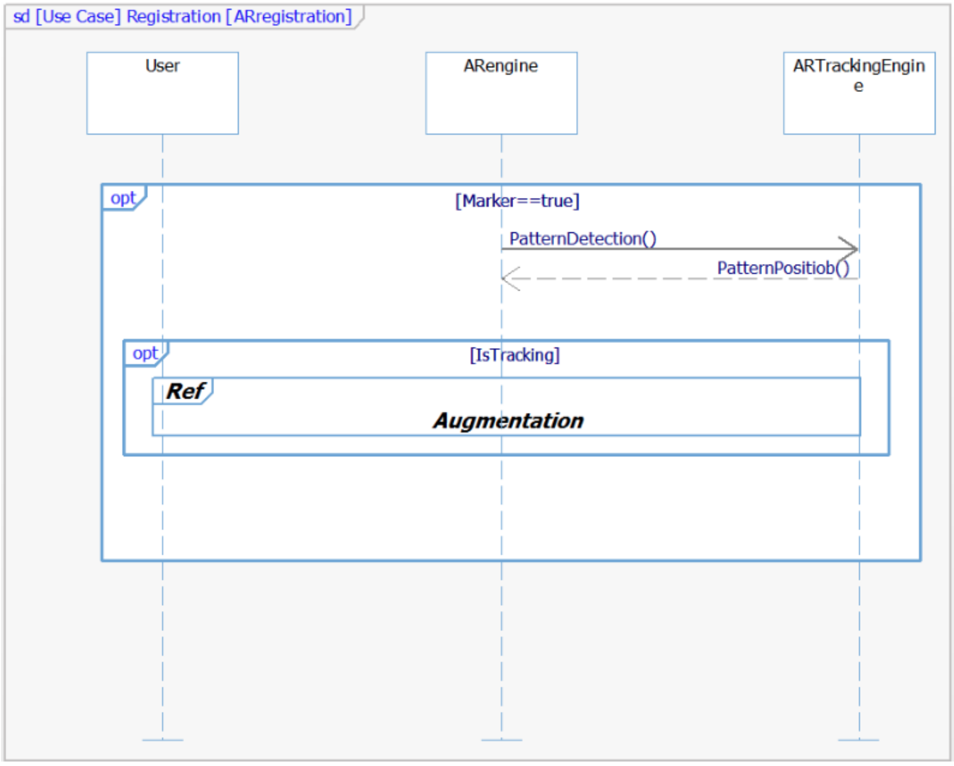


Figure E.2: AR Registration Sequential Diagram

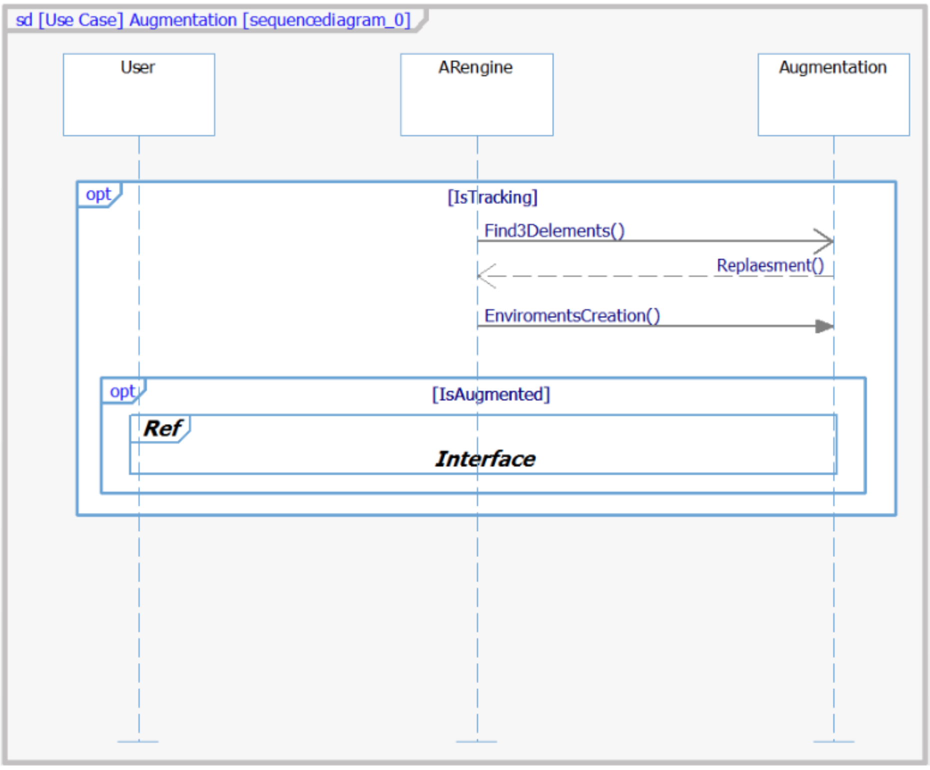


Figure E.3: AR Augmentation Sequential Diagram

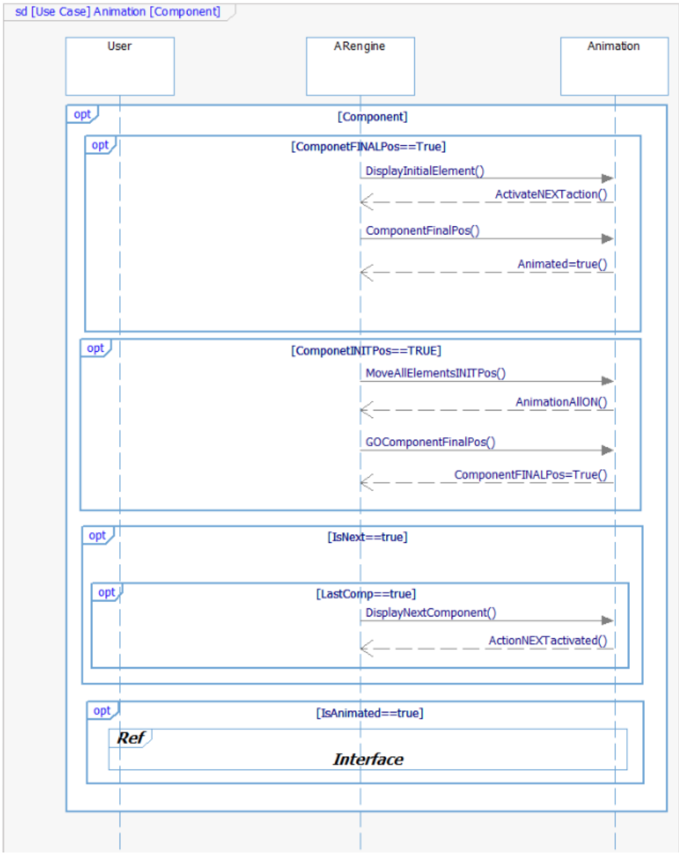


Figure E.4: Animation Sequential Diagram

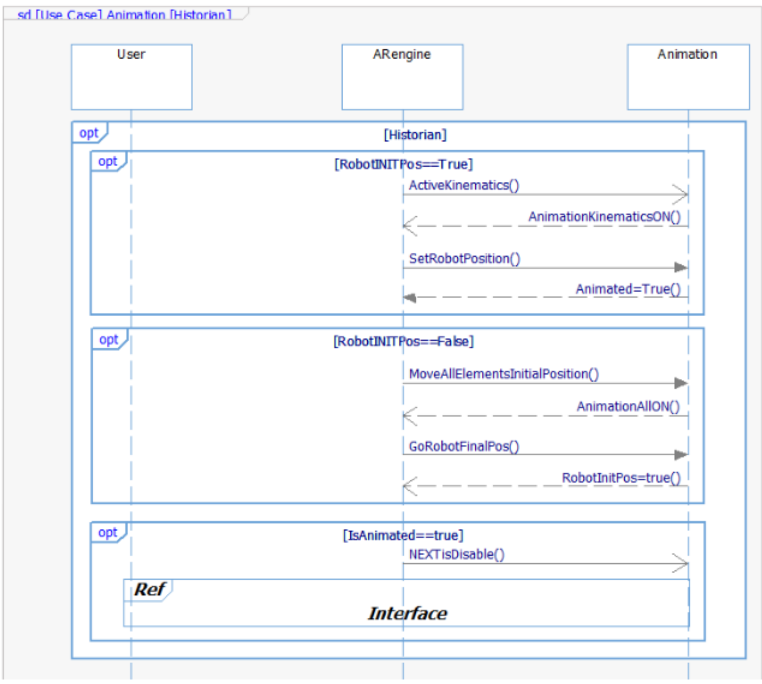


Figure E.5: Historian Action Sequential Diagram

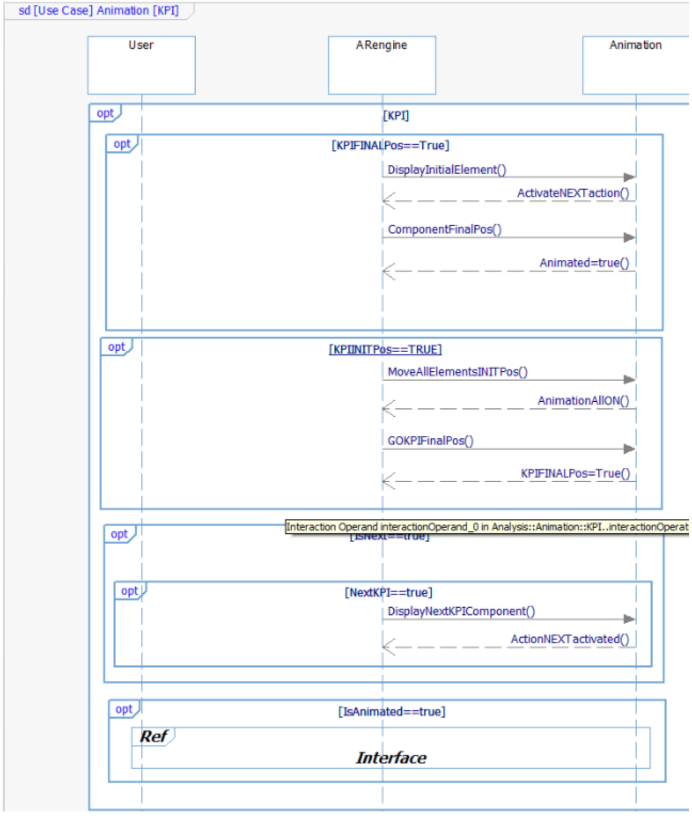


Figure E.6: KPI Action Sequential Diagram

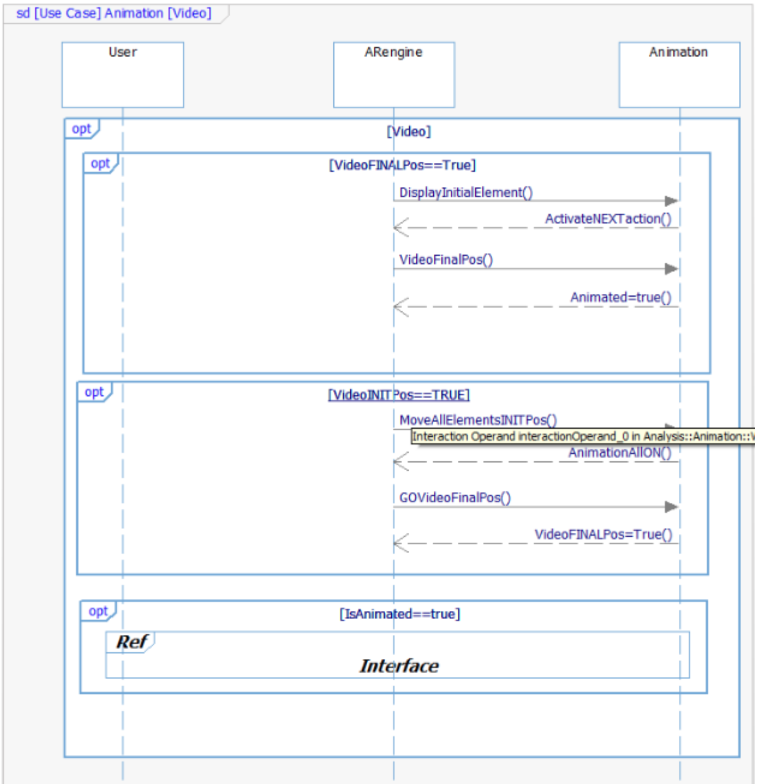


Figure E.7: Video Action Sequential Diagram

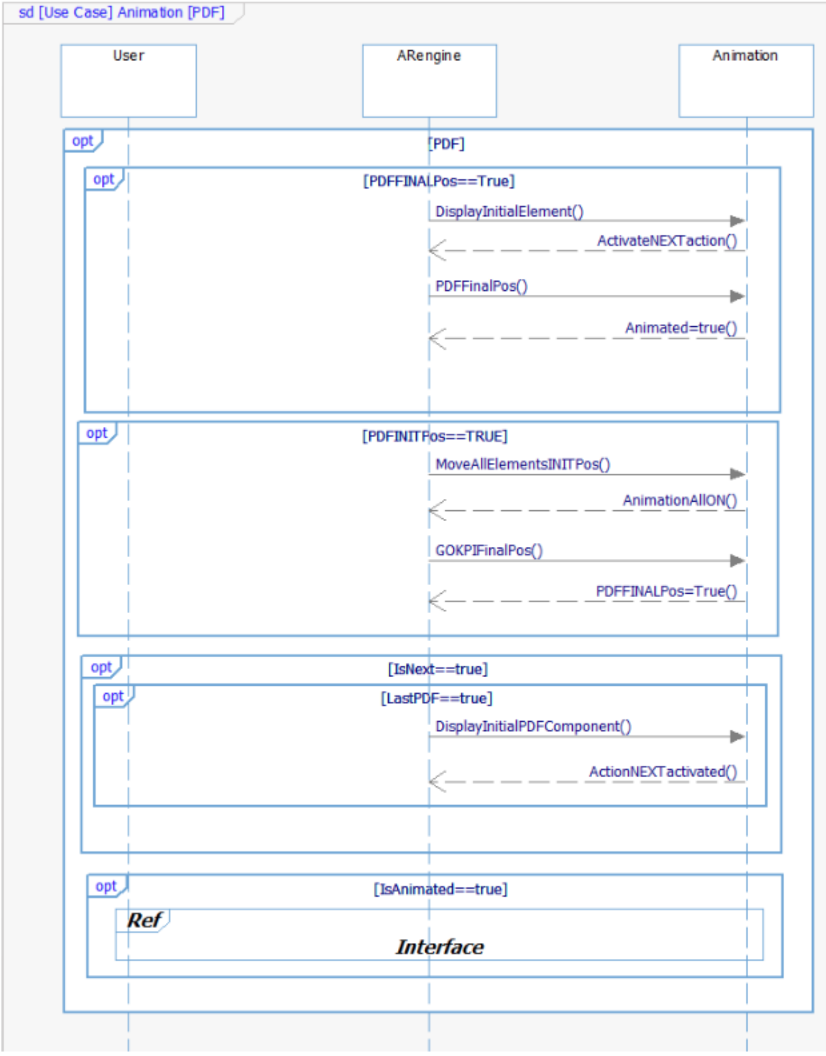


Figure E.8: PDF Action Sequential Diagram