Alma Mater Studiorum – Università di Bologna
Dipartimento di Informatica – Scienza e Ingegneria (DISI)

Dottorato di Ricerca in
Computer Science and Engineering

Ciclo XXXI

Settore Concorsuale di Afferenza: 09/H1
Settore Scientifico Disciplinare: ING-INF/05

AUGMENTED WORLDS: A PROPOSAL FOR MODELLING AND ENGINEERING PERVASIVE MIXED REALITY SMART ENVIRONMENTS

Candidato
Dott. Ing. ANGELO CROATTI

Coordinatore Dottorato
Prof. PAOLO CIACCIA

Supervisore
Prof. ALESSANDRO RICCI

Tutor
Prof. MIRKO VIROLI

Esame Finale Anno 2019

to whom unexpectedly coloured my life
`#meglioindue`

Acknowledgements

I want to thank first Prof. Alessandro Ricci for his supervision during my PhD: his support has provided an invaluable contribution to both my professional and personal growth. A special thank to Prof. Mirko Viroli and Dr. Silvia Mirri for the support and suggestions provided as members of my internal evaluation board. A particular mention is for Prof. Abraham "Abey" Campbell and Prof. Rem Collier, for hosting me in Dublin during one of the most exciting experiences of my life so far. I must also thank Prof. Gregory O'Hare and Prof. Carlos Carrascosa, reviewers of this dissertation, whose valuable comments helped to improve my work. A special thank to colleagues and friends with whom I spent my PhD years at the *Pervasive Software Lab* in Cesena, especially Sara Montagna and Michele Braccini. Last, but not least, I owe everything I am to my family and my sincerest thanks are for them: without their remarkable support and their continual encouragement I would never have achieved this outstanding goal.

Angelo Croatti – February 15$^{\text{th}}$, 2019

Abstract

In recent years, the remarkable technological advancement has revolutionised the ICT panorama offering nowadays the opportunity to exploit several technological convergences to reduce the gulf existing between the physical and the digital matter, between the physical real world and every computational software system or application. Along with Pervasive Computing – entered in the mainstream with the concept of Internet of Things (IoT) – Mixed Reality (MR) is going to be an essential ingredient for the design and development of next future smart environments. In particular, in such environments is feasible to imagine that the computation will drive the augmentation of the physical space, and software will also be executed in a cyber-physical world, eventually populated with of (interactive) holograms.

After an initial exploration of the state of the art about augmentation technologies both for humans and the environment, in this dissertation we present the vision of *Augmented Worlds* (AW), a conceptual and a practical proposal for modelling and engineering next future pervasive mixed reality smart environments as distribute, multi-user and cooperative complex software systems. On the one hand, a meta-model is formalised and opportunely discussed to offer to the literature a conceptual tool for modelling AWs. On the other hand, also a concrete infrastructure – called MiRAgE – is designed and developed to produce a platform for engineering and deploy such innovative smart environments.

The work carried out in this dissertation fits into the scientific literature and research of Pervasive Computing and Mixed Reality fields. Furthermore, part of the contribution is related also to the area of Cognitive Agents and Multi-Agent Systems, due to the AWs orientation to be deeply connected to a layer involving autonomous agents able to observe and act proactively in the smart environment.

Keywords — *Augmentation, Mixed Reality, Pervasive Computing, Smart Environments, Agents, MiRAgE*

Index

Introduction

Within recent years, the remarkable technological advancement that has strongly improved and expanded the worldwide ICT panorama offers nowadays plenty technologies that may be exploitable to reduce the existing gulf between the physical and the digital matter, between what is real and what is virtual. Among others, one of the challenges that new technologies allow dealing with concerns the definition and the engineering of new *smart physical-digital environments* able to offer more and more support and assistance to people's everyday life. In particular, support and assistance mediated by the software and the computation: that is, by the availability of smart technologies and, in other words, by what improperly people define *artificial intelligence*.

An essential ingredient for the design and development of next future smart environments can be found in Pervasive/Ubiquitous Computing, entered in the mainstream with the concept of *Internet of Things* (IoT) [2]. According to late 1980s, Mark Weiser vision – the idea of smart environments comes from the original definition of Ubiquitous Computing that promotes the ideas of "*a physical world that is richly and invisibly interwoven with sensors, actuators, displays, and computational elements, embedded seamlessly in the everyday objects of our lives, and connected through a continuous network*" [65].

Nowadays, the definition of smart environments has evolved a lot from the original vision, according to new technological opportunities and researches in the pervasive computing wide area. At the beginning of the third millennium, a more comforting vision for humans defines a smart environment as "*a small world where different kinds of smart devices are continuously working to make inhabitants' lives more comfortable*" [15]. In other words, this vision suggests that we are at the beginning of a "new era" allowing us to conceive a novel kind of smart environments to improve the quality of humans' work and life. A scenario where the computation is pervasively coupled with the real environment and where technology can proactively act considering the real-time context of the people who live, work and cooperate into the real world.

Nevertheless, if we want to move towards this new era of hybrid environments where the *digital matter* is deeply coupled with the physical one – in addition to IoT and Pervasive Computing – we must also consider another primary ingredient. In particular, we refer to Mixed and Augmented Reality and to related technologies able to engage the user in an immersive and visual perception of the digital entities, staying anyway deeply connected to reality. For instance, thanks to the availability of new ad-hoc wearable visors and smart glasses, Mixed and Augmented Reality techniques are going to become even more used to proposes new interaction strategies for innovative software systems dedicated to particular purposes – especially in the industrial context, but not only that. Also, they allow rethinking on how exploiting Augmented Reality is could be possible to refactor existing software solutions, improving them with new features and opportunities.

The vision promoted by this dissertation for conceiving new smart physical-digital smart software-based environments is rooted to the idea of integrating Pervasive Computing with Mixed Reality technologies, in order to propose a new conceptual framework to shaping a new category of smart and proactive software systems.

In other words, it is feasible to assert that times are now ripe to think about how to conceive new smart environments that arise from the convergence of those technologies that can *augment* both the physical environment and humans themselves. That is, on how to use the technologies to produce systems offering a seamless integration between the real and the digital to augment the physical reality to provide new features, not for humans only but also for (not necessarily computational) elements or entities composing the reality (e.g., object, tools, embedded devices, and so on).

The literature, however, seems to have not been faced or discussed yet the problem of proposing a shared and integrated vision that could lead to a reduction of the abstraction gap existing between nowadays available enabling technologies and the hybrid systems that will constitute with high probability the reality of the next future. On the one hand, it is an open problem, whose solution assumes a general value from the research point of view: finding a solution for modelling and engineering these software systems – referring in particular on the cooperative and collaborative aspect among human users themselves and among also them and the environment – could represent a relevant research result. On the other hand, it is clear that the availability of such a study is not only purely theoretical but can immediately be applied to develop software solutions in specific contexts, if supported by a proper software platform offering the right level of abstraction.

Considering what stated so far, this dissertation wants to deal with the challenge to reduce the existing *distance* that currently exists between the digital and the physical worlds, investigating on how to propose a general-purpose software solution to model and engineering these novel *cyber-physical systems* – here referred with the name of *Augmented Worlds*. Avoiding to entering into details, we can simply define Augmented Worlds as complex software systems where is significant the strong coupling between the physical layer and the digital one; where autonomous and proactive computational entities offer human users the possibility to exploit smart capabilities for acting, interacting and cooperating according to their dynamic, real-time and distributed context. The dissertation will provide to the reader a more specific definition and further details about it.

Dissertation objectives

Despite the exhaustive presentation of the motivations, objectives and also the contribution of this dissertation will be provided in Chapter 4 – that is, after having proposed to the reader a gradual path spanning from a discussion about augmentation technologies and the background useful concepts towards the investigation of state of the art looking for related works – it is anyway appropriate anticipating here the macro-goals of this dissertation.

The primary motivation that has brought to results proposed and discussed here is dealing with the not yet appropriately encountered in literature critical challenge of incorporating pervasive computing with mixed reality techniques to offer new software solutions able to support future smart environments requirements. In particular, this dissertation follows a research work devoted to:

- proposing a conceptual meta-model, capturing and defining main concepts for the design of innovative smart environments integrating pervasive computing with mixed reality (the "*augmented worlds*" vision);
- integrating Cognitive Agents and Multi-Agent Systems (MAS) with the proposed meta-model to define a concrete model to design agent-oriented pervasive mixed reality software solutions where computational components can proactively act and reason according to humans considered as first-class abstractions for the model;
- build a prototype of a software platform enabling the engineering and the development of these complex agent-based pervasive mixed reality systems, with the purpose to facilitate programmers' work offering them a higher abstraction layer if compared to currently available frameworks.

Summarizing, the goal of this dissertation is to propose a model for augmented worlds with a general worth, able to become a conceptual reference to be adopted for the design of such complex software systems. Furthermore, the related software platform – called MiRAgE – should act as a concrete starting point for a reference platform to develop real and running instances of augmented worlds, to be used in specific context and domains.

As a concluding note, it is very important to clarify to the reader that this work mainly wants to open a research perspective for the design of augmented worlds. In other words, this is only a first step towards the here proposed vision of opportunely coupling real and computational ecosystems. As the title of this dissertation suggests, is presented here a "*proposal*" and not a theory or a formal answer to a particular open issue. The author well knows that a lot of work must be carried out to continue the research in this direction. Nevertheless, we firmly believe that this work can offer many ideas and starting points for further discussions and comparisons of future activities in the same research area. In the end, also this can be considered as another (collateral) objective of this dissertation.

Structure of the dissertation

The dissertation proposed in this volume follows a path split into four macro parts. Following paragraphs describe the content of each part referring to involved chapters.

PART I – Background and Motivations This part is mainly devoted to explore and analyse technological context that composes the background for the dissertation. Chapter 1 propose an overview of augmentation technologies, focusing on those devoted to augmenting both humans and the physical environment. Besides, Chapters 2 and 3 offer an overview, respectively, on Mixed Reality and Pervasive Computing, that represent the two central ICT macro areas to which the dissertation refers. Finally, Chapter 4 concludes the introductory part with a discussion on motivations, challenges and objectives of the dissertation. This is the chapter where the contribution of the dissertation is clarified, presenting as well an exhaustive exploration of state of the art and the literature to discuss the relevant related works.

PART II – Augmented Worlds This second part gets to the heart of the dissertation, discussing the vision and concepts that bring to the definition of a conceptual meta-model for designing Augmented Worlds, meant as Pervasive Mixed Reality oriented smart environments. Firstly, Chapter 5 gives a formal definition of what an augmented world is, discussing related functional features and relevant challenges to address. According to this definition, the chapter continues exploring details of the first class elements composing the augmented world meta-model (including augmented entities, holograms, agents, human users and so on). Therefore, Chapter 6 describes a few relevant examples of application domains to discuss in practice the effectiveness of the proposed meta-model.

PART III – The MiRAgE Infrastructure After facing the conceptual/formal side of the dissertation in the previous part, here we move to the most engineering side of the dissertation. Chapter 7 discuss the design of MiRAgE, a software platform on the augmented worlds meta-model to enabling an adequate development of pervasive mixed reality smart environments, also supporting their deployment, their simulation and testing. The developed MiRAgE prototype – whose building details are reported in the chapter – offers tools, software libraries and a runtime to allow developers to design and run augmented worlds over an adequate technological abstraction layer. In particular, it allows to get without any effort a distributed and shared environment as the base for each augmented world instance, accessed by multiple human users that can also act cooperatively in the Mixed Reality environment along with proactive agents. Consequently, Chapter 8 addresses the issue of the evaluation of the MiRAgE platform, considering the main requirements for augmented worlds previously discussed and proposing also real case studies with the purpose to point out the strengths and limitations of the framework.

PART IV – Conclusions and Future Works The dissertation ends with this final part where, in Chapter 9, a conclusive discussion on the proposed work is dealt with. In particular, the chapter reports an overall evaluation and analysis on limitations and critical points of the proposed approach for modelling and engineering Augmented Worlds. The last section of the chapter explores the road ahead in terms of future works and proposes a research agenda with some possible food for thought and other related studies.

Part I

Background and Motivations

Chapter 1

Augmentation Technologies

> *"Advances in augmentation technologies offer improvements to human health, quality of life and functional performance."* — Mike Daily et al., 2017 [21]

In recent years augmentation technologies witnessed an impressive development and progress, making them ready for being used out of research labs. Although the term "*augmentation technologies*" refers to a broader set of visions, in this dissertation we refer to the subset related to modern technologies able to enriching/augmenting the physical world (and reality) with some digital/computational functionalities and services. These technologies can have a significant impact from an application point view, enhancing human cognitive capabilities and allowing for rethinking the way in which people work, interact and collaborate both in the physical and in the digital space. If wearable computing and mixed reality are a direct example of augmentation technologies for humans, ubiquitous/pervasive computing – entered in the mainstream with the name of Internet of Things (IoT) – provides a further and related form of augmentation of the physical world. In particular, in this latter case, the augmentation is given by open ecosystems of connected and invisible computing devices embedded in physical objects, spread in the physical environment and equipped with different kinds of sensors and actuators. From a conceptual level, the fruitful integration of these augmentation technologies can be considered as the keystone to conceive novel kind of smart environments, designing and developing a broad class of new software systems and applications.

1.1 Human Augmentation

In 1968, the computer science pioneer Douglas C. Engelbart introduced his vision of an "augmentation system" as the first research objective of the new Augmented Human Intellect (AHI) Research Centre at Stanford Research Institute [26]. The principles and techniques for designing Engelbart's augmentation system had to consider issues not

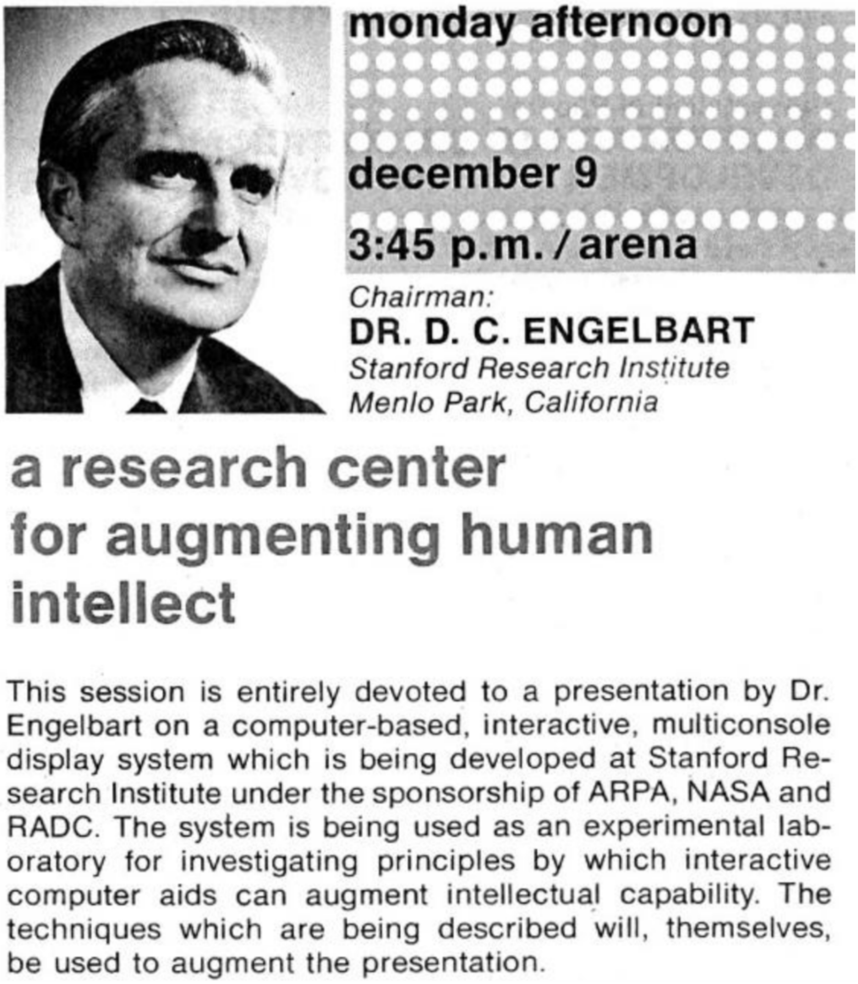


Figure 1.1: The Engelbart original presentation flyer (`www.1968demo.org`).

only for the technology but also for changes both in ways of conceptualising, visualising and organising working material and in procedures and methods for working individually and cooperatively. The Engelbart framework for augmenting the human intellect refers to *increasing the capability of a man to approach a complex problem situation, to gain comprehension to suit his particular needs and to derive solutions to problems* [25]. In particular, Engelbart's vision does not refer to a class of systems able to produce particular suggestions for particular situations. Conversely, it considers a context (a "way of life") where *the human feel for a situation usefully co-exist with powerful concepts, streamlined terminology and notation, sophisticated methods, and high-powered electronic aids.*

Nowadays – fifty years later than Engelbart main work – hardware advances in the

ICT panorama allow us to rethink to human augmentation building new software solutions within this broad and still open research context in augmenting humans capability enhancing their ways of living, working and interaction.

1.1.1 An extension of the self

Human augmentation is a term that can be generally used to refer to technologies that enhance human productivity or capability. Within the wide category of technologies allowing for a kind of human augmentation, we can make some different classifications.

A set of technologies that is relevant to the notion of human augmentation considered in this dissertation involve hardware technologies related to mobile and wearable computing supporting different degrees of mixed reality — the leading example is given by smart glasses like, i.e., Google Glass, Epson Moverio, Vuzix Smart Glasses, Daqri Smart Helmet, and so on. Conceptually, this kind of devices allows extending people cognitive capabilities – improving the way in which they perform their tasks and collaborate – becoming an "*extension of the self*" [63].

The maturity of these technologies is witnessed by the products and solutions that are gradually entering into the mainstream. Among the others, a main recent one is Microsoft HoloLens, that generates a multidimensional image visible to the user wearing the holographic helmet so that he or she perceives holographic objects in the physical world. Another example of advanced and mature technology in this field is represented by Meta2 visor, that easily allows users to visualise, touch, grab, and pull digital content.

1.1.2 Software-based human augmentation

As remarked in [71], it is not only a matter of a hardware augmentation: the software plays a key role here such that we can talk about *software-based human augmentation*. The level of sophistication achieved by the techniques developed in the context of Mixed Reality (MR) research – supported by the availability of more and more powerful hardware (sensors, processors, displays, wearables – makes it possible today to focus on the design of mixed reality-oriented software systems, assuming that as the basic enabling functionalities. Although some issues are still open, is very clear how the technology for the Mixed Reality has evolved from the very first prototype proposed in by I. Sutherland in 1965 of a visor able to increase the visual perspective of humans (called "The ultimate Display", see Figure 1.2).

Nowadays, Mixed Reality oriented technology is improved a lot if compared to first device realisation attempts, it has been perfected and also made available to the mainstream out of laboratories. Nevertheless, the purpose of this category of devices is still the same: augmenting humans projecting them towards new worlds where digital and physical are strongly coupled and where new features are offered to them for targeting new lifestyles.

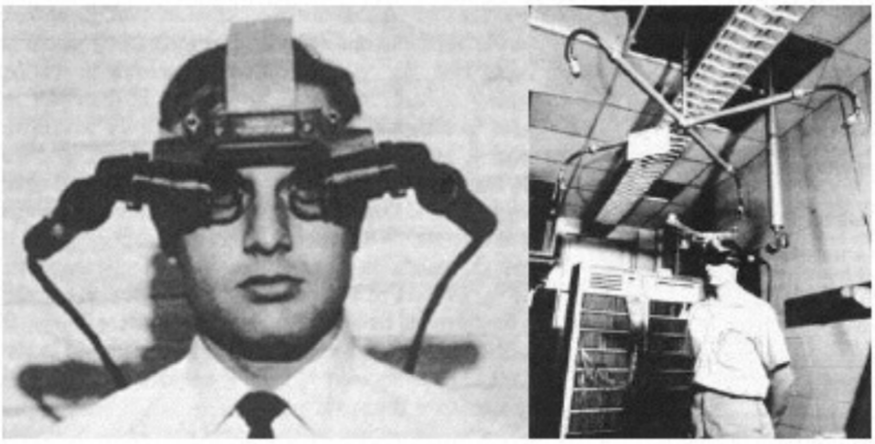


Figure 1.2: The Sutherland's *Ultimate Diplay*, aka "*The Sword of Damocles*".

1.2 Environment Augmentation

To conceive novel smart environments, besides the human augmentation technologies – augmenting physical world inhabitants – we cannot forget the physical world itself: also an *environment augmentation* is thus required.

1.2.1 Cyber-Physical Systems

Talking about environment augmentation, nowadays, with the term Cyber-Physical Systems (CPS) we define a class of systems that deeply embed cyber/digital capabilities in the physical world, either on humans, infrastructure or platform with the aim to transform interactions with the physical world [46]. For a more detailed definition of what a CPS is, authors of the *Ptolemy Project*[a] developed at Berkeley University have proposed a conceptual map helping for better comprehension.

Beyond conceptual definitions, to switch toward the complete identification of CPS is also required to refer to contexts and technologies able to bring the computation directly into the environment.

The vision of pervasive/ubiquitous computing and IoT [64, 56, 36, 48] provides elements for devising the augmentation of the physical world. In this case, the augmentation is given by ecosystems of connected computational embedded devices spread in the physical environment. Data generated by these devices are typically collected and managed in the cloud – as big amount of data – and accessed by mobile/cloud applications. This embedded computing layer augments the functionalities of the physical things that people use every day and, again, are going to have a substantial impact on how people work, interact and collaborate.

[a]`https://ptolemy.berkeley.edu/projects/cps/`

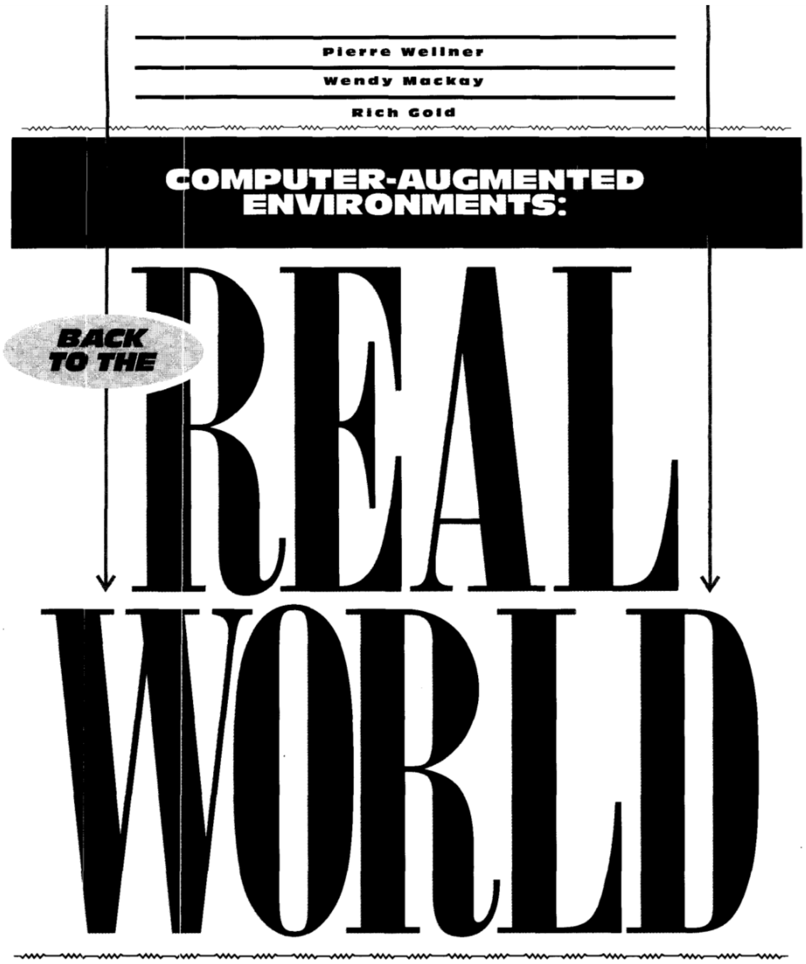


Figure 1.3: The historical cover of the 1993 Special Issue on Computer-Augmented Environments [67] published by the Communication of ACM journal.

1.2.2 The Computer Augmented Environment vision

This view about environment augmentation strongly recalls the idea of computer-augmented environments by P. Wellner et al. [67], in which the digital/cyber world is merged with the physical one. In 1993, he claimed that "*the future of computing will emerge from the introduction of computers to augment objects of the physical world*". Moreover, "*although ubiquitous computing and augmented/mixed reality are very different technologies*" – Wellner and his colleagues wrote – "*they are united in a common philosophy: the primacy of the physical world and the construction of appropriate tools that enhance our daily activities.*" Wellner concludes his paper with a hope: he hoped that his work could "*excite the imagination of the computer science community, exiting from apparent satisfaction given by the desktop embodiment of the computation*". Nowadays we can undoubtedly assert that the IoT and Mixed Reality represent the main actualisation of Wellner's hope.

1.3 The role of Context

Talking about augmentation emerges the exigence to take into account the notion of *context*. Augmenting humans means proposing to them specific computational functionalities according to the current context in which each human is immersed in (e.g., according to him/her current location, current health status, other humans nearby, and so on). Vice versa, the augmentation of the environment takes in account current context to update the environment itself in real-time (e.g., in an augmented home interior artificial lights can be managed according to external context and weather). Furthermore, the use of context turns out to be very important when we consider interactive software systems where the system's status evolves rapidly and in real-time with the physical environment and actions of its inhabitants.

At the end of the twentieth century, Schilit and Theimer firstly introduced the idea of context [57]. In their work, they refer to a notion of context user-centric as a set of information composed by user location, user nearby people and objects and state and changes of these objects. Other definitions added to this set further elements, among which temporal-informations, orientation, gaze direction and even the user emotional state [11, 23]. A more comprehensive definition of context has been proposed by Dey and Abowd in [1], where the context is defined as:

"*any information that can be used to characterize the situation of an entity*" where "*an entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves*".

Considering the given definition, it is clear how the context plays a key role in the augmentation technologies: a promising and effective augmentation requires taking in high consideration the reference context of the involved "entity". So, we can assert that a system with the purpose to provide augmentation, exploiting related technologies, has to be also considered as a context-aware software system. Referring again to [1], a system is context-aware if

"*it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task*"

implying a system that should be able to adapt according to context.

Regardless of the techniques used to build it, an essential aspect of an augmentation system emerges from the following observation: if we want to feel the gap between the real and the virtual, we have to make these two distant layers – the physical and the digital one – more "homogeneous" exploiting software to enrich the entire environment for seamless integration. To achieve this result, contextual information plays a fundamental role: using lots of context data, the augmentation system become more and more powerful concerning the effectiveness of functionalities proposed to users. Nevertheless, because the amount of collected contextual data can increase very quickly, in the design process of a context-aware augmentation system is very important to do not forget dealing with performances and computational coast.

1.4 On the adoption of Augmentation Technologies

The even wider availability of new devices, supporting both humans and environment augmentation, demonstrates how ICT industries last years have invested in developing such devices to make them available to be technically used in large scale.

Now, the main related question is if people are ready for massive adoption of this set of technology or, in other words, if people are ready to change their way of living and working consistently. Avoiding entering in ethical/philosophical issues, this section is limited to investigating if this new technological direction can be taken into account and when.

The initial euphoria for augmentation technologies industrialisation has been dramatically reduced by the Google Project Glass attempt: launched with huge mass adoption expectations, after a little period, Google Glass has been retired from the consumer market. Despite this, ICT industry has not stopped production of augmentation technologies and smart devices. The diffusion process of these technologies merely has gone through the typical Gartner hype cycle[b]. After an initial peak of inflated expectations and some adoption in niche applications, these devices have not shown mass adoption yet. Anyway, perspectives in this direction are good, and the initial phase of the hype cycle has come to its close: the second and third generation of products have been funded and developed which are appealing to even more customers. Numbers give a proof of this affirmation: e.g., in 2018 the demand for Mixed Reality oriented devices has increased a lot if compared to previous years[c]. Moreover, many industrial domains are planning yet a re-engineering of their internal processes considering the rise of this new era for the IT sector, mainly based on the concept of augmentation.

Admittedly, we have a long way ahead in the direction of mass adoption of these technologies. However, the current scenario is undoubtedly a remarkable context for new research horizons that assume the availability of augmentation technologies as one of their key starting points.

[b]`https://www.gartner.com/en/research/methodologies/gartner-hype-cycle`
[c]`https://www.idc.com/getdoc.jsp?containerId=prUS43639318`

1.5 Augmentation in Industry 4.0

Concluding this concise analysis about augmentation technologies both for humans and the physical environment we cannot avoid to consider the emerging phenomenon in the ICT context we are witnessing, commonly defined as the fourth industrial revolution. Better known as *Industry 4.0*, this “revolution” is marked by emerging technology breakthroughs in various fields such as robotics, artificial intelligence, nano and biotechnology, autonomous vehicles, the Internet of Things and Augmented Reality. Figure 1.4 depicts the main “ingredients” of Industry 4.0.

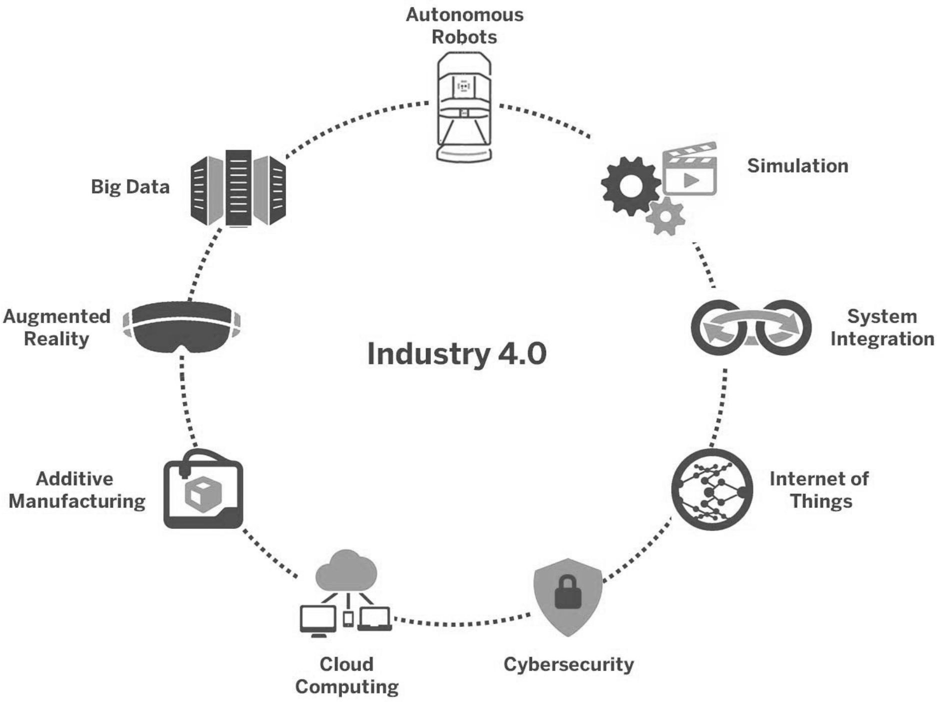


Figure 1.4: Industry 4.0 main “ingredients”. (this figure has been taken from `aethon.com/industry-4-0-means-manufacturers`)

An interesting aspect of Industry 4.0 related to the development and the adoption of augmentation technologies is the concept of *digital twin*. Considering a physical object – e.g. industrial machinery – or a business process, a digital twin can be defined as an evolving digital profile of the historical and current behaviour that helps to optimise business performances [43]. More technically, the real power of a digital twin is that it can provide a comprehensive real-time connection connection between the physical and digital worlds, of course in the industry context but is not limited to it. The connection should be available both for physical objects and for humans – e.g. both machinery and workers of a particular factory. In this direction, a digital twin can be considered as the augmentation vision encouraged by Industry 4.0 for factories and companies of the third millennium.

The augmentation promoted by the digital twin concept is wider if we consider the integration of augmentation technologies in the process. Applications are not only limited to get feedbacks from machinery and workers of a factory to be processed in real-time by business algorithms. We can do much more: we can alter the production process in real-time, and we can modify the way in which workers execute their actions providing them contextual information and a continuous training considering, e.g., the real environment as a test bed. The digital twin can allow companies to have a complete digital real-time and dynamic image of their production sectors and may enable them, e.g., to solve physical issues faster by detecting them sooner.

Chapter 2

Mixed Reality: an overview

> *"The future of human consciousness will be a hybrid affair. We will live and work in a ubiquitous environment, where physical reality and a pervasive digital layer mix seamlessly according to the logic of software. This is Mixed Reality, and it will soon simply be reality!"* — John Rousseau, 2016 [55]

This chapter proposes an overview of Mixed Reality concepts. The analysis performed here is not meant to be an exhaustive and detailed study about Mixed Reality techniques or technologies but is limited to explore foundations of this context useful to support the dissertation.

2.1 Definitions

Mixed Reality (MR) generally refers to the merging of real and virtual worlds to produce new environments and visualisations where physical and digital objects coexist and interact in real time [58]. As defined by P. Milgram and F. Kishino in the early 90s, it is "*anywhere between the extrema of the virtuality continuum*" [39], that extends from the completely real through to the completely virtual environment with augmented reality and augmented virtuality ranging between (see Figure 2.1). In other words, Mixed reality offers a way to seamlessly blending digital content (in this dissertation we will generally refer to this digital content with the term *holograms*) into the physical world so which appears realistic or natural [44].

The Virtual Continuum spectrum allows distinguishing among several degrees of augmentation that bring a human user to experience the Mixed Reality, moving from the real environment to the virtual one. In particular:

- *Physical Reality* (PR), is the world where people live and work, that is, what all people consider the "real world".

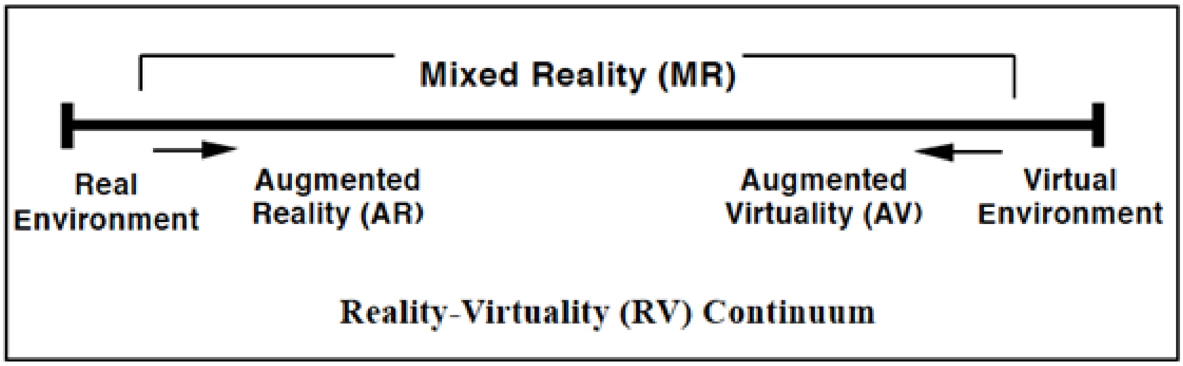


Figure 2.1: The P. Milgram and F. Kishino Virtual Continuum [39].

- *Augmented Reality* (AR), allows to layering digital content mainly over the physical world. The main difference with MR is that in AR we can assume the digital content overlays the real world, rather than integrate deeply within.
- *Mixed Reality* (MR), blends holograms seamlessly into the physical world. The key aspect of MR is the ability to stay grounded in the real world, as opposed to being completely immersed in a digital space.
- *Virtual Reality* (VR), brings the human user to a completely digital space where he/she is immersed within a digital experience without references to the real world.

Except for the PR (assuming its meaning is well known) and to avoid confusion, Figure 2.2 proposes a concise but effective definition on how in this dissertation distinguishes among AR, MR and VR. Unlike a VR experience that completely immerses the users in a fully digital world, MR keeps users anchored to the real world by blending holograms among the physical objects users see around them. Mixed Reality is an augmentation of the real spaces we live in because the physical space is always visible and most important in sync with users perspective.

2.2 Holograms in Mixed Reality

As announced in the previous section, in this dissertation with the term holograms we generally refer to the Mixed Reality digital content. There is not a formal definition of the concept of *hologram*, but we can assume than holograms are *digital representations of anything we can imagine blended into our physical spaces*. Holograms can have any shape and, possibly, make any sound. Also, holograms can modify the physical space enriching it. Nowadays, to see and interact with holograms, users need to wear proper devices like smart-glasses and holographic visors (i.e., Microsoft HoloLens or Meta2). Exploiting

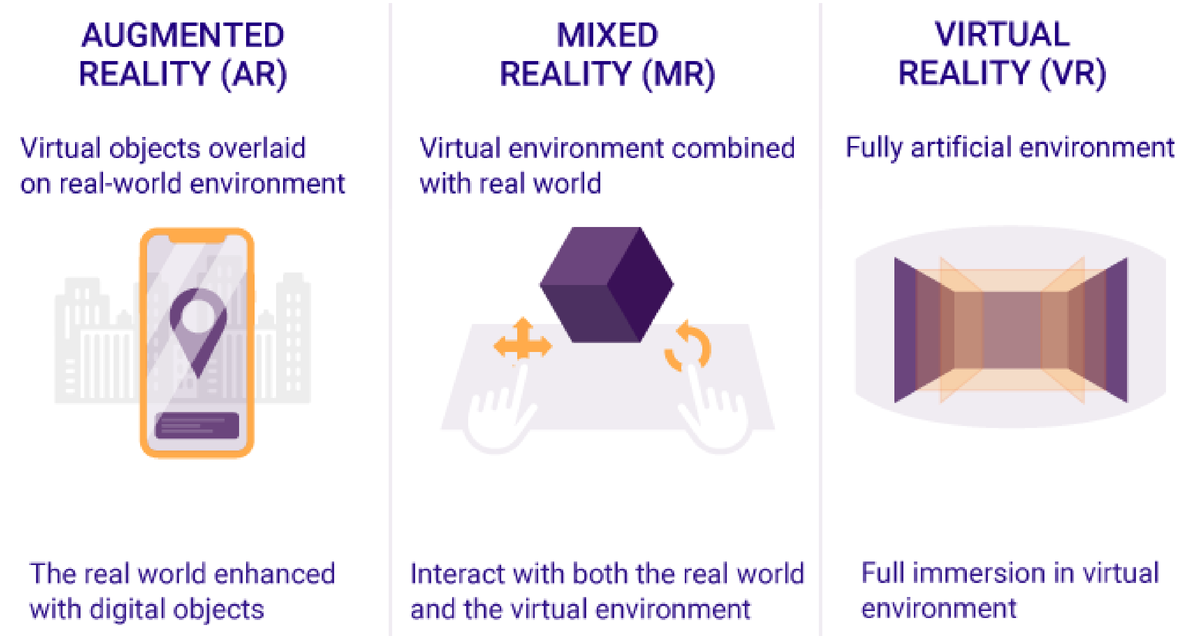


Figure 2.2: The accepted definition in this dissertation. This figure has been adapted from the original taken at `https://rubygarage.org/blog/difference-between-ar-vr-mr`

such visors, users can see holograms according to their perspective, touch/grab them and possibly (like in the case of the Magic Leap One visor) exploiting multiple focal planes to experience normal eyes focus like in the real world. In other words, if holograms are well designed, in the short next future humans could interact with holograms like real things, sometimes even without immediately understanding what is real and what is digital.

2.2.1 Challenges

Although nowadays MR and, generally, the interaction with holograms represent a rather interesting research field, it is a context that is still expanding. Even though research on Mixed Reality started several decades ago, only a few years ago arisen real exploitable technologies offering a compelling experience for users. Therefore, still today, no shortage of challenges.

High Expectations and Paradigm Shift. Expectations around Mixed Reality and Holograms are high. Many movies (like *Iron Man* by J. Favreau, to name one) led people to build a very futuristic vision regarding Mixed Reality and its related potentialities. Software tools and hardware for Mixed Reality are far from those visions: we are only at the beginning of the road towards them. Furthermore, designing and developing MR experiences is such a different paradigm than all the other forms of computing that we know. We need to redefine a lot of design abstractions and interaction elements/mechanism to build novel Mixed Reality environments revolutionising people's way of working and living. Surely, this new paradigm shift will provide many interesting new opportunities along with a significant number of challenges.

Spatial Thinking. Designing Mixed Reality environments requires the software engineer, the designer and the developer to think in a three-dimensional space. The software must go out of the traditional 2D screen (of a PC or a mobile device), the physical 3D space is the stage within which build digital entities. Nevertheless, thinking spatially rather than screen-based is undoubtedly harder, and this represents a significant challenge for MR environments designers. So far, designing a user experience meant thinking about elements and concepts to be used on a screen (like windows, menus, icons, dialogs, buttons and so on). The Mixed Reality experience brings many other issues to the forefront for software designers. For example, how to model holograms and how to place them according to the real world. How to propose an experience with holograms like real objects. How to design interactions and how to propose 2D elements, i.e. like text messages to users in a 3D environment. And so on.

High computational capabilities. To propose an effective Mixed Reality experience high computational capabilities are required. Visors and devices need to process a huge amount of data acquired from sensors to model the physical space and place holograms according to the user's perspective, current field-of-view and movements. Unfortunately, this requirement does not sit well with the fact that Mixed Reality devices should be comfortable and lightweight to wear. Currently available visors whereas they are incredibly performable generally requires to be connected by wire to a computer (like Meta2). Otherwise, if they have all sensor embedded and they need high computational capabilities, they generally sensibly reduce the Field-of-View that the user can exploit to observe the Mixed Reality (like Microsoft HoloLens).

Real People. Except for the interaction dimension, in software design humans can be generally considered as external actors of the system. In Mixed Reality, instead, real people must become first-class entities to be taken into account when designing and developing a related software system. Holograms could consider people's current gaze and location to change their shape. Moreover, functionalities proposed could depend on the real people's real-time "status". This surely represents an exciting challenge to deal with.

2.3 Building Mixed Reality

Many people use the term Mixed Reality interchangeably with Augmented Reality. Considering the given definitions, we can assume that MR defines a category of applications broader than AR. Generally speaking, we can assume that all AR applications are also MR applications while not all MR applications can be considered AR ones.

Nevertheless, Augmented Reality plays a key role, representing indeed the most important ingredient involved in the process of building a Mixed Reality oriented environment. R. Azuma has proposed the first commonly accepted definition of Augmented Reality in 1997, which listed the three main features of AR [3]. In particular:

1. combining real and virtual realities;

2. real-time interaction;

3. registration of digital elements in the three-dimensional real-world reference system.

More recently, A.B. Craig [16] refined the original definition given for AR, asserting that:

augmented reality is a medium in which digital information is overlaid on the physical world that is in both spatial and temporal registration with the physical world, and that is interactive in real-time.

Entering in details, the key aspects of AR, and so of MR, are discussed in the following two paragraphs.

Superimposition of the Digital Content to the Physical World. In AR, the computer-generated digital content is superimposed on (a view of) the real world without occluding to the user the perception of the physical reality. The digital content could range from simple and static 2D elements to complete dynamic 3D graphics models and renderings. According to the features of the device used to *see* the AR, digital content can be merged with the real world in different ways. Different experiences can be obtained using wearable see-through devices – like smart-glasses or visors – where the user is immediately able to observe the mixed reality rather than using 2D screens – like monitors or smartphones/tablets – where the physical reality is back-propagated to the screen using a proper camera.

Registration with the Real World. Spatial registration is a crucial element if we want to produce a Mixed Reality experience. A digital element needs to have physical space or location in the real world regardless of the position of the user observing it. If the perspective of the user changes according to user's movements in the real worlds, a digital element – assuming that a 3D hologram currently represents it – should keeps its position but proposing a different "aspect" according to its geometry. In a sense, holograms must be managed as physical objects at least for what concerns the registration issue. For this reason, the registration with the real world must be both spatial and temporal. Unfortunately, temporal registration is even more difficult to achieve due to the inherent time lags involved with processing data from sensors observing and mapping in real-time the physical world. Because the view of each hologram depends on the user's point of view, the hologram rendering must be updated every time the user changes its position/perspective. In case the user changes his/her perspective very rapidly, the lag in the computing might provide a noticeable lag in the perceived scene. Moreover, the spatial registration can be either *absolute* – e.g., according to a GPS-based (Global Positioning System) location – or relative to another real or digital entity.

2.3.1 Augmented Reality Techniques

Every time that an AR-based software system wants to update the view of the user on the Mixed Reality environment to show him/her holograms, two steps are requested to be performed. First of all, the system needs to identify the current state of the physical world and compute state of the digital world accordingly. Secondly, emerge the exigence to display holograms in registration with reality in an effective way considering human perceptions.

Generally, the real-time state of the physical world is obtained exploiting sensors provided by the user device, in particular cameras and computer vision techniques are used for this purpose.

A significant issue in augmented reality is "*knowing where the user is, respect to the real-world reference system*" that is, dealing with motion tracking and location. In augmented reality, determine the user perspective means computing the camera's *pose* that is represented by a position in the 3D space with six-degrees of freedom (6DOF). The pose refers to the position and orientation of the camera. To identify camera's pose with reasonable precision inertial sensors (like accelerometer or gyroscope) and location sensors (like compass or GPS) can be used.

Once the AR system detects the camera's pose – that is when the AR system knows the current location and orientation of the user – holograms and the digital content can be registered with the real world and proposed to users in a particular instant of time according to his/her perspective.

According to the literature, several techniques can be considered, possibly combined, to determine where the user is, and which is his/her view perspective. The most basic technique – known as *Marker-based AR* – uses fiducial markers to enable tracking in order to determine the device's pose. Beside markers, also real-time visual tracking techniques can be applied. In particular, the reference is to those techniques allowing for a continuous mapping of the real world exploiting a large set of detected fiducial points – also known as *Markerless AR*.

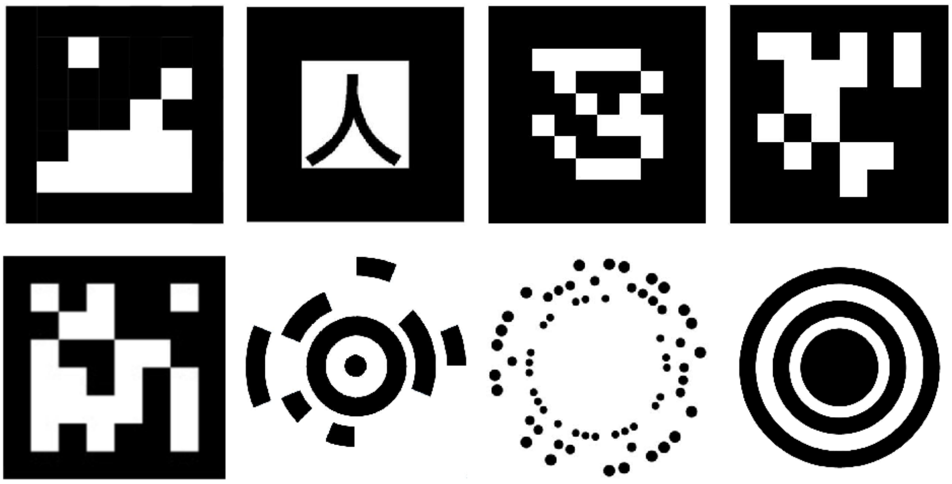


Figure 2.3: Example of AR markers.

Marker-based AR

The most known computer vision technique for registering digital elements in AR is based on the usage of *fiducial markers*. Commonly, a marker is – but is not limited to – an asymmetrical 2D image representing a very simple pattern with the aim to be quickly recognisable by computer vision algorithms (typically square, black and white patterns). Because of their predefined shape and pattern, markers are easily detectable and located within the current camera obtained frame. So, they are used for a quick pose calculation. Furthermore, the high contrast of black and white squares help for easier detection.

Markers are placed in the real environments with the purpose of identifying a reference system to be used for coupling digital layer with the real one. Using a marker placed in a well-known position of the real world the tracking function uses the camera to estimate the pose of the device in real time based on what it is "seeing".

Markerless AR

In poor lighting conditions, when the camera is far away from the tracked marker, or when the camera focus cannot be well established, fiducial marker detection techniques based on natural feature point detection can be used as combined or standalone approaches. Natural Feature Tracking (NFT) is an AR technique for recognising and tracking a scene to determine the camera's pose, deliberately avoiding to use markers. NFT can use items that are embedded in the real world to enhance tracking points and regions within the current user's view. The result of such form of tracking can be called markerless tracking because the "markers" are not visible to the user.

Although markerless approaches could appear the best solution, note that tracking the camera pose in unknown environments can be a challenge. In order to reduce this

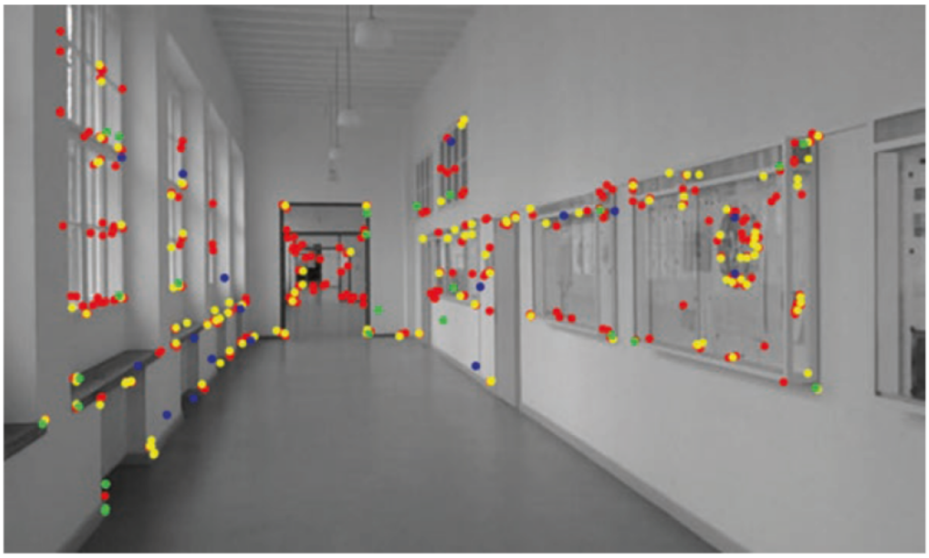


Figure 2.4: An example of markerless AR feature points tracking.

challenge, a technology known as SLAM (Simultaneous Localization and Mapping), has been developed during the last years which redefines the original idea of markerless AR based on feature points. Nevertheless, the NFT can be less computationally expensive than a full SLAM system and generally is more suitable for mobile devices.

SLAM starts with an unknown and empty environment where the augmented reality device tries to generate a map and localise itself within the map. Through a series of computationally expensive algorithms, SLAM uses the IMU (inertial measurement unit) sensor data to construct a map of the unknown environment while using it at the same time to identify where it is located.

Finally, markerless AR typically also uses the GPS feature to track user location in outdoor scenarios combined with visual tracking to obtain related orientation.

2.3.2 The interaction issue

One of the critical features of AR is that it is, by its nature, an interactive medium. If the digital content must be perceived as a part of the real world, as people can interact with physical objects also in AR systems users should be able to interact with holograms similarly. In fact, interaction plays a key role in the overall user experience.

The concept expressed by the term *interaction* can assume much meaning, especially in the real world. Nevertheless, in an AR environment, the interaction is generally limited to action to be performed on digital elements, e.g., gazing or grabbing them, moving from an original position to another position and manipulating holograms geometries.

Interaction in AR is a challenging aspect that mostly depends on the device used to track and show holograms in the Mixed Reality environment. The ability to interact with holograms proportionally increases and become more powerful in relation to the efficiency of the used AR device in terms of own sensors and computational capabilities.

2.3.3 Multi-users experience

Although most current AR-based applications propose single-user experiences – as the Real World is intrinsically a multi-user cooperative environment – Mixed Reality environments become more interesting when also the digital layer and holograms can be shared among multiple users and interacted simultaneously.

Proposing a cooperative multi-user experience means having a shared digital layer in real-time. First of all, users should be able to observe the same holograms but from their own perspective. In this case, using markers, this feature could be simply obtained – e.g., with two users that simultaneously observe the same marker that proposes the same reference systems and orientation for both of them. Unfortunately, this is a solution that hides the real issue: in this case, the state of the hologram is not really shared among users. For instance, if a user moves the hologram, the other user will continue to see the hologram in the original position. To solve the proposed issues there is the exigence for both users to refer to the same instance of the hologram currently under observation from both — a kind of centralisation of the digital layer instance, accessed by multiple users

Not surprises that to experiment a full cooperative/collaborative multi-user MR environment emerge needs that are far to be dealt with by currently available software infrastructural solutions. Especially when the cooperation is not only intended among humans but also involves autonomous software entities and other physical things belonging to the real world.

2.4 Hardware and Software for Mixed Reality

To complete and enrich the overview about Mixed Reality, in this section a brief exploration of the technological panorama in terms of devices and software supporting MR is proposed. The objective is to give a taste of both devices and software tools currently on the forefront.

2.4.1 Mixed Reality Enabling Devices

One of the most important parts of augmented reality is the ability of the user to see his or her environment. Literally, to *see* the Mixed Reality, a user of a mixed reality based systems has to be adequately equipped with devices to allow the uses to observe (and interact with) holograms and the digital content overlapped to the physical reality. Also,

(a) Microsoft HoloLens. (b) Meta2 by Meta.

Figure 2.5: Examples of Mixed Reality enabling devices.

the augmented reality device has to “see” the physical environment, and that involves a computer-based vision system. Thus, a camera combined with a display is an appealing configuration for such a device.

Among others, for the purposes of this dissertation, categories of enabling devices of interest are mainly two:

- *Hand-held devices*, like smartphones and tablets;

- *Wearable see-through devices*, like smartglasses and MR visors.

Assuming the features of smartphones and tablets nowadays are well-known to everyone, and because that that category of devices does not belong to the wearables category – in fact, they do not allow to users hands-free experiences for effectively interacting with holograms – in the following we will focus on the second of these categories. In particular, we will consider two of the most advanced devices currently available: HoloLens[a] by Microsoft and Meta2[b] by Meta.

Microsoft HoloLens

Microsoft HoloLens is a holographic computer produced by Microsoft, running Windows 10 operative system. It is completely untethered no wires, phones, or connection to a PC needed (see Figure 2.5a).

It allows to pin holograms in the physical environment and provides a new way to see the world exploiting markerless AR techniques. Microsoft HoloLens features see-through, holographic, high-definition lenses and spatial sound so the user can see and hear holograms in the world around him/her. Complete with advanced sensors and a new Holographic Processing Unit (HPU) that understands the world around the user,

[a]`https://www.microsoft.com/en-IE/hololens`
[b]`https://www.metavision.com/`

Microsoft HoloLens can run without any wires while processing terabytes of data from the sensors in real-time. The HoloLens projected screen moves as the user moves his/her head. The user is also able to control apps either with voice commands or by using the equivalent of a mouse click, the air tap. The holograms the user will see with Microsoft HoloLens can appear life-like and can move, be shaped, and change according to interaction with users or the physical environment in which they are visible.

In the front of the visor, there are most of the sensors and related hardware, including the cameras and processors. The HoloLens features an inertial measurement unit (IMU) (which includes an accelerometer, gyroscope, and a magnetometer) four "environment understanding" sensors (two on each side), an energy-efficient depth camera with a 120x120 degrees angle of view, a 2.4-megapixel photographic video camera, a four-microphone array, and an ambient light sensor. HoloLens features IEEE 802.11ac Wi-Fi and Bluetooth 4.1 Low Energy (LE) wireless connectivity.

Meta2 – Meta Development Kit 2

The MR visor Meta2 is an Head-Mounted Display (HMD) developed by the American Company Meta (see Figure 2.5b). It is a wired display that requires to be connected to a proper PC to work, to propose to the user the holograms and MR digital content. The set of sensors whose Meta2 is equipped with includes: depth sensor, 720 megapixel camera, IMU technologies, microphones and speakers. It allows a SLAM-based (Simultaneous Localization And Mapping) environment reconstruction, gestures recognition and gaze tracking. Finally, it is characterised by a horizontal Field-Of-View (FOV) of 90 degrees, that is significantly wider than any other competitor (e.g. Microsoft HoloLens has a FOV of only 30 degrees).

2.4.2 Libraries and tools supporting Augmented Reality

Several companies are offering toolkits and libraries to aid developers in creating applications exploiting augmented reality. A few of the nowadays most popular tools are described in this section. In particular, we are going to consider ARToolkit[c] by DAQRI, Vuforia[d] by PTC Inc., ARCore[e] by Google and ARKit[f] by Apple.

DAQRI ARToolkit

Developed at the Nara Institute of Science and Technology in Japan and first released in 1999, ARToolkit represents probably the most known software library to build

[c] https://github.com/artoolkit
[d] https://www.vuforia.com/
[e] https://developers.google.com/ar/
[f] https://developer.apple.com/arkit/

augmented reality based applications. Based on OpenCV[g], this library uses computer vision algorithms to solve the problem of tracking the user's viewpoint. The ARToolkit video tracking libraries calculate the real camera position and orientation relative to physical markers in real time. This enables the easy development of a wide range of Augmented Reality applications. Some of the features of ARToolkit include:

- single-camera or stereo-camera (camera position/orientation tracking);
- tracking of simple black squares (any square marker patterns);
- tracking of planar images (natural feature markers);
- camera calibration, optical stereo calibration, square marker generation, and natural feature marker generation utilities;
- fast enough for real-time AR applications.

ARToolkit has been acquired by the well-known AR American company DAQRI and is currently used to develop software for DAQRI wearable technology products, manly designed for industries.

PTC Inc. Vuforia

Firstly developed as a standalone library for creating and managing augmented reality but now fully integrated into the Unity game engine, Vuforia is one of the best-known available augmented reality tool-sets. Vuforia enables 3D content to be placed in physical environments exploiting the Vuforia Engine, which provides computer vision functionality to recognise objects and reconstruct environments. It uses such computer vision technology to recognise and track planar images (Image Targets) and simple 3D objects in real time. This image registration capability enables developers to position and orient virtual objects, such as 3D models and other media, in relation to real-world images when they are viewed through the camera of a mobile device.

The Vuforia SDK supports a variety of 2D and 3D target types, 3D Multi-Target configurations, and a form of addressable Fiducial Marker, known as a VuMark. Additional features of the SDK include localised Occlusion Detection using "Virtual Buttons", runtime image target selection, and the ability to create and reconfigure target sets programmatically at runtime. Vuforia's 3D reconstruction stack – called Smart Terrain – offers to developers to access to the surfaces and objects found in the environment. With Smart Terrain, developers can create AR applications that can interact with physical objects in the world. Vuforia provides APIs in C++, Java, Objective-C, so AR applications developed using this library are therefore compatible with a broad range of mobile devices, especially those running Android O.S. or Apple iOS.

[g]`https://opencv.org/`

Google **ARCore**

ARCore is the innovative Google's platform for building augmented reality experiences. It enables a mobile device (in particular but not limited to those running the Google Android O.S.) to sense its environment, understand the world and interact with the obtained information. ARCore uses three key capabilities to integrate virtual content with the real world as seen through your phone's camera. In particular:

- **Motion tracking**, to understand and track the device position relative to the world. ARCore detects visually distinct feature points and uses them to compute its change in location. The visual information is combined with inertial measurements from the device's IMU to estimate the pose (position and orientation) of the camera relative to the world over time.

- **Environmental understanding**, allowing the mobile device to detect the size and location of all type of surfaces: horizontal, vertical and angled surfaces like the ground or walls. ARCore is constantly improving its understanding of the real-world environment by detecting feature points and planes. ARCore looks for clusters of feature points that appear to lie on common horizontal or vertical surfaces, and makes these surfaces available to the developer within the concept of *plane*.

- **Light estimation**, allowing the mobile device to estimate the environment's current lighting conditions. ARCore provides to the application the average intensity and color correction of a given camera image. This information lets the programmer light virtual objects under the same conditions as the environment around them, increasing the sense of realism.

ARCore uses the concept of *anchor* to ensure that the application will be able to track an object's position over time. Moreover, it allows the usage of Cloud Anchors to create collaborative or multi-player applications, sharing the environment and digital content.

Apple **ARKit**

The Apple's response to Google ARCore is called ARKit and is a developer's toolset and platform for creating digital objects that blend in with the real world. ARKit uses the device's camera, sensors, and custom software to match and integrate these AR objects quite realistically. In particular, it combines device motion tracking, camera scene capture, advanced scene processing, and display conveniences to simplify the task of building an AR experience.

Unfortunately ARKit is only available for Apple devices with the Apple A9 and later processors. This constraint, justified by Apple by the fact that "*only processors of the A9 class deliver breakthrough performance that enables fast scene understanding and lets programmers build detailed and compelling virtual content on top of real-world scenes*", practically has limited a lot the diffusion and the adoption of ARKit. In fact, developers seem to prefer ARCore by Google, that is suitable also on iOS devices.

2.4.3 Game Engines to develop MR applications

By definition, a game engine is a software development environment designed for people to build video games. The core functionality typically provided by a game engine includes a rendering engine ("renderer") for 2D or 3D graphics, a physics engine or collision detection (and collision response), sound, scripting, animation, artificial intelligence, networking, streaming, memory management, threading, localization support, scene graph, and may include video support for cinematics.

Despite this, although they were born for a completely different purpose, nowadays game engines represent the primary medium to develop Mixed Reality software systems. In particular, the most used to this purpose is the well-known game engine Unity[h] by Unity Technologies, that is used as the entry level by almost all AR toolkits and libraries and support the deployment of applications for almost all mobile devices. Its main competitor, Unreal Engine[i] by Epic Games – although it is very powerful for developing games – is still less used for developing Mixed Reality software systems.

Unity

Without question, the most useful all-around tool for developing mixed reality applications is Unity. This constantly improving tool allows designers and developers to create, compose, and test out their mixed reality ideas with real models and code within an integrated development environment (IDE).

Basically Unity gives users the ability to create games in both 2D and 3D, and the engine offers a primary scripting API in C#, for both the Unity editor in the form of plugins, and games themselves, as well as drag and drop functionality. Within 2D games, Unity allows importation of sprites and an advanced 2D world renderer. For 3D games, Unity allows specification of texture compression, mipmaps, and resolution settings for each platform that the game engine supports, and provides support for bump mapping, reflection mapping, parallax mapping, screen space ambient occlusion (SSAO), dynamic shadows using shadow maps, render-to-texture and full-screen post-processing effects.

Related to the development of MR software systems, Unity support the native integration of the Vuforia library. Also, ARCore can be used within Unity for the same purpose.

[h] https://unity3d.com/unity
[i] https://www.unrealengine.com/

Chapter 3

Pervasive Computing : an overview

"The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it." — Mark Weiser, 1999 [64]

This chapter proposes an overview of Pervasive Computing and related concepts as a background for the topic of this dissertation. However in scientific literature a subtle difference between the terms "pervasive" and "ubiquitous" related to the computation can be identified, for purposes of this dissertation the two terms will be used indiscriminately, preferring the "pervasive" one.

3.1 Introduction

The paradigm of pervasive computing describes ubiquitous computing environments that provide anytime and anywhere access to information services while making the presence of the system invisible to the user. It represents a growing trend of embedding computational capability into everyday objects to make them effectively communicate and perform useful tasks in a way that minimises the end user's need to interact with computers as computers. Pervasive computing devices are network-connected and continuously available.

Historically, Pervasive Computing was envisioned by Mark Weiser in late 1970s and emerged at the conjunction of research and development in many areas which include embedded and devices and systems, wireless communications, and distributed, mobile and context-aware computing. Nowadays the primary goal of Pervasive Computing is to make devices "smart", thus creating a sensor network capable of collecting, processing and sending data, and, ultimately, communicating as a means to adapt to the data's context and activity. In essence, this novel form of computation is about a network that can understand its surroundings and improve the human experience and quality of life.

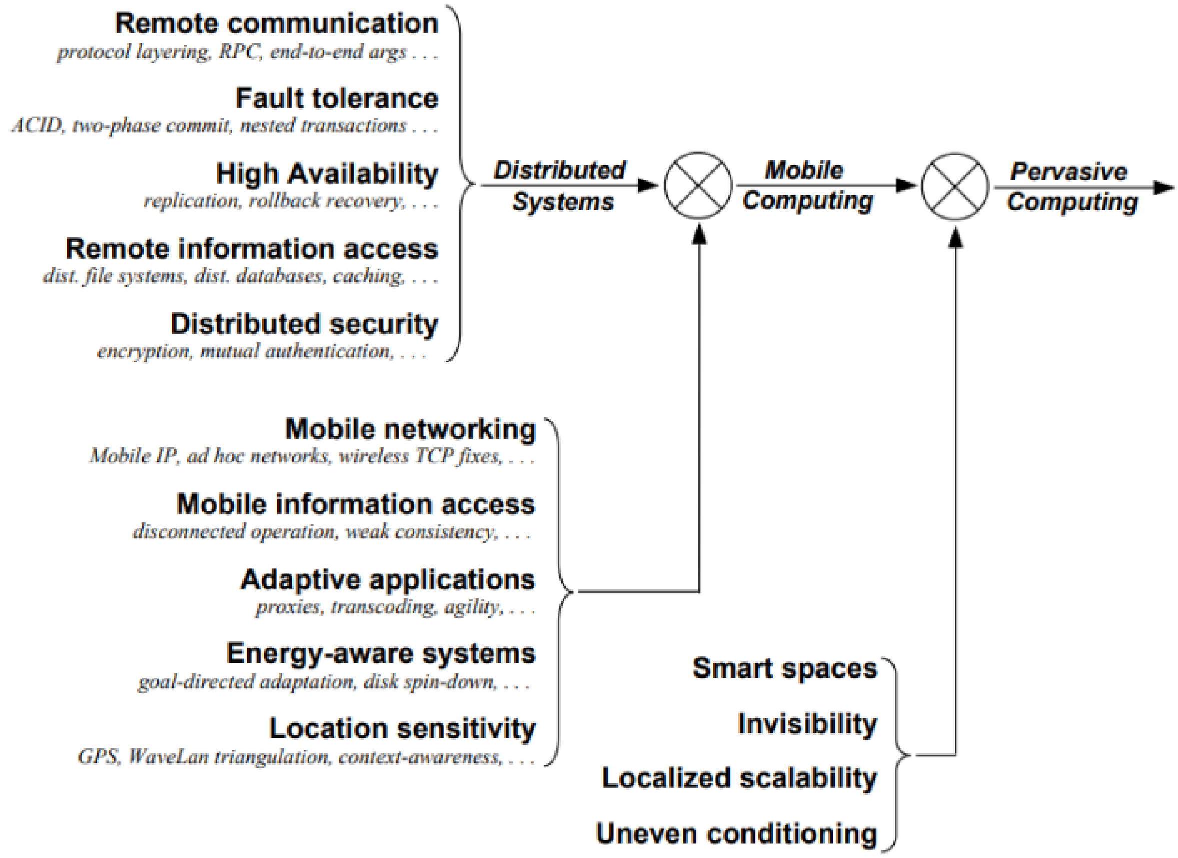


Figure 3.1: A taxonomy of issues from Distributed to Pervasive Computing [56].

Generally speaking, Pervasive Computing represents a major evolutionary step in a line of work where at the begin we can find the area of Distributed Computing passing through the area of Mobile Computing. To better clarify features and problems of such research line, Figure 3.1 taken from [56] reports a taxonomy of issues bringing the computation from distributed computing field to the pervasive one. As reported by authors of the figure, it is important to note that it describes logical relationships, not temporal ones. Although the evolution of research effort over time has loosely followed this picture, there have been cases where research effort on some aspect of pervasive computing began relatively early. For example, work on smart spaces began in the early 1990s and proceeded relatively independently from work in mobile computing.

3.1.1 Distributed Computing

Distributed computing is a field of computer science that studies distributed systems. Generally speaking, a distributed system is a system whose components are located on different networked computers, which then communicate and coordinate their actions generally through a message passing mechanism. In particular, a more formal definition for a distributed system is: *a collection of autonomous computing elements that appears to its users as a single coherent system.* This definition refers to two characteristic features of distributed systems.

The first one is that a distributed system is a collection of computing elements each being able to behave independently of each other. A computing element – a node of the system – can be either a hardware device or a software process. As a consequence of dealing with independent nodes, each one will have its notion of time. In other words, we cannot assume that there is something like a global clock. This lack of a common reference of time leads to fundamental questions regarding the synchronisation and coordination within a distributed system.

A second element is that system's users (both people and applications) believe they are dealing with a single system. More specifically, in a single coherent system, the collection of nodes as a whole operates the same, no matter where, when, and how the interaction between a user and the system takes place. For a distributed system, offering a single coherent view is often challenging: in literature this feature is called *distribution transparency*.

For what concerns pervasive computing, distributed systems expose foundational features like:

- *remote communication*, including protocol layering, remote procedure call, asynchronous communication, the use of timeouts, and so on;
- *fault tolerance*, including atomic transactions, distributed and nested transactions, and so on;
- *high availability*, including optimistic and pessimistic replica control, mirrored execution, optimistic recovery;
- *remote information access*, including caching, function shipping, distributed file systems, and distributed databases;
- *security*, including encryption-based mutual authentication and privacy.

Cloud Computing

Nowadays, a particular "instance" of distributed computing is called Cloud Computing, that is a shared pool of configurable computer system resources and higher-level services that can be rapidly provisioned with minimal management effort, often over the Internet. Cloud computing adopts concepts from Service-oriented Architectures (SOA). In particular, in the field of cloud computing we can generally distinguish among several models of providing resources "as a service":

- *Infrastructure as a service* (IaaS), refers to online services that provide high-level APIs used to dereference various low-level details of underlying network infrastructure like physical computing resources, location, data partitioning, scaling, security, backup etc.

- *Platform as a service* (PaaS), is a category of cloud computing services that provides a platform allowing customers to develop, run, and manage applications without the complexity of building and maintaining the infrastructure typically associated with developing and launching an app.

- *Software as a service* (SaaS), is a software licensing and delivery model in which software is licensed on a subscription basis and is centrally hosted. SaaS applications are also known as Web-based software, on-demand software and hosted software.

Beyond these specifications, which would require a more thorough study, nowadays cloud computing represents the foundation for supporting pervasive computing. In particular, whereas on the one hand, the pervasive vision of computation is about a kind of computation spread in the physical environment, on the other hand, a so seamless computation requires a reliable infrastructure where demanding expensive computation or the storage and the elaboration of the vast amount of data produced by pervasive computing.

3.1.2 Mobile Computing

From a theoretical point of view, it would be possible to talk about pervasive computing without considering the mobile component. In this way, however, fundamental features that nowadays define and complete the ICT area related to pervasive computing would be lost.

Mobile computing generally refers to a set of IT technologies, products, services, and operational strategies and procedures that enable end users to gain access to computation, information, and related resources and capabilities while mobile.

Historically, mobile computing arises from the idea to support the philosophy of the computation *anytime* and *anywhere*. Despite this, the evolution of mobile computing requested several decades to reach the actual level of development. Avoiding to entering in details about this evolution, today, we are assisting to the definition of digital ecosystems composed by mobile devices and mobile computing applications. The technical capabilities of our mobile devices have improved significantly to the point where factors such as screen real estate, input capabilities, processing power, network speed, and battery lifetime are much less of an issue than only half a decade ago. Mobile devices are becoming more and more important and widespread, they will soon be the dominating point of access to the Internet. Importantly, what we are witnessing here is not just the development of even smarter smartphones with improved abilities to imitate desktop PCs in miniature. It is a radical evolution of a major computing platform for new applications allowing us to do things that could not be done before.

In other words, we are going towards the direction to create digital ecosystems in which mobile computing plays a central role in concert with other ubiquitous computing resources.

Wearable Computing

Concluding this section about mobile computing we want to consider a particular area related to what is commonly called Wearable Computing.

Wearable Computing generally refers to those wearable systems designed to be used within a broad spectrum of mobile contexts where the user must interact simultaneously with the environment and with the system. Wearable computers may be worn under, over, or in clothing, or may also be themselves clothes.

An important distinction between wearable computers and mobile ones is that the goal of wearable computing is to position or contextualise the computer in such a way that the human and computer are seamlessly intertwined. Important features of Wearable computing, in addition to mobile computing features, are:

- the ability to interact effectively with the environment exploiting appropriate sensors, which can provide a degree of knowledge both related to the user's activities and to the context that surrounds him/her;
- the direct interaction of the system with the user should be minimal and generally should not involving hands, preferring a hands-free interaction mechanism;
- the system should be strongly context-aware and should act considering both user and environment context.

3.1.3 Ubiquitous Computing

After the brief discussion about both Distributed and Mobile Computing as precursors of what nowadays we call pervasive/ubiquitous computing – referring back to Figure 3.1 – we want to conclude the definition of what pervasive computing is, discussing now those (macro)features shaping pervasive computing beyond mobile and distributed software systems.

Effective use of Smart Space. By embedding computing infrastructure in physical spaces, a smart space brings together two worlds that have been disjoint until now. The fusion of these worlds enables sensing and control of one world by the other. In other words, a pervasive computing system involves not only mobile devices managed by users but also the entire physical space where users are immersed in.

Invisibility. Each computational entity involved in the pervasive computing system must not distract users requiring a direct interaction when possible. This means that if a pervasive computing environment continuously meets user expectations and rarely presents him with requests for interactions, it allows the user to interact almost at a "subconscious level".

Localized Scalability. Scalability for pervasive computing system is a challenging issue. In particular, is required to have a way to adapt the system to scale according to direct/indirect interactions of users with the systems in a particular area of the pervasive environment. In other words, it is not required for the system to have a uniform degree of scalability for all of its parts. Conversely, it is necessary, instead, the ability to support good scalability for those parts most solicited considering actions that users perform within the system.

3.2 The Internet of Things (IoT)

The Internet of Things (IoT) [38] has mostly evolved out of pervasive computing. Though some argue there is little or no difference, IoT is likely more in line with pervasive computing rather than Weiser's original view of ubiquitous computing. Prof. Kevin Ashton is accredited for using the term "Internet of Things" for the first time in 1999 [2]. He believes the "things" aspect of the way we interact and live within the physical world that surrounds us needs serious reconsideration, due to advances in computing, Internet, and data-generation rate by smart devices.

The broad definition of the Internet of Things vision is a world where the internet is much more than the collection of multimedia content as it is today: it extends into the physical, real-time world using a myriad of small or even tiny computers. A more formal definition for the IoT is given in [33] where is argued that:

the Internet of Things is a system of physical objects that can be discovered, monitored, controlled, or interacted with by electronic devices that communicate over various networking interfaces and eventually can be connected to the wider internet.

A "smart thing" is a physical object that's digitally augmented with one or more of features among *sensors*, *actuators*, *computation* and *communication interfaces*. Things extend the world we live in by enabling a whole new range of applications, including industrial applications, emergency management, healthcare, demotics, etc.

An essential role of IoT is to build a collaborative system that is capable of adequately responding to an event captured via sensors, by effective discovery of crowds and also successful communication of information across discovered crowds of different domains. IoT is also recognized by the impact on quality of life due to its features enabling for:

- expanding the communication channel between objects by providing a more integrated communication environment;
- facilitating the automation and control process, whereby administrators can manage each object's status via remote consoles;
- saving in the overall cost of implementation, deployment, and maintenance, by providing detailed measurements and the ability to check the status of devices remotely.

Concluding, we can observe that what IoT tries to picture is a unified network of smart objects and human beings responsible for operating them, who are capable of universally and ubiquitously communicating with each other.

3.2.1 Architecture

Several research groups have proposed reference architectures for IoT. For purposes of this dissertation, the main interesting architectural proposal for building the IoT is the Service Oriented Architecture (SOA) and, in particular, a SOA where each service is built as a RESTful Service.

Service Oriented Architecture (SOA)

A service-oriented architecture (SOA) is a style of software design where services are provided to the other components by application components, through a communication protocol over a network. A service is a discrete unit of functionality that can be accessed remotely and acted upon and updated independently. In SOA, services use protocols that describe how they pass and parse messages using description metadata.

Nowadays, SOA can be implemented with Web services. This is done to make the functional building-blocks accessible over standard Internet protocols that are independent of platforms and programming languages. These services can represent either new applications or just wrappers around existing legacy systems to make them network-enabled.

Within the IoT field, SOA ensures the interoperability among the heterogeneous devices. Generally, in such architecture a complex system is divided into subsystems that are loosely coupled and can be reused later (modular decomposability feature), hence providing an easy way to maintain the whole system by taking care of its own components.

RESTful Service Design

An extension to SOA is represented by API-Oriented Architectures, that basically is an approach to build SOAs using Web APIs and Representational State Transfer

(REST) [28], as defined by R. Fielding in his PhD Thesis in 2000. Generally speaking, REST is defined as an architectural style, which means that it is not a concrete systems architecture, but instead a set of constraints that are applied when designing a systems architecture. More formally (from [28]):

REST provides a set of architectural constraints that, when applied as a whole, emphasizes scalability of component interactions, generality of interfaces, independent deployment of components, and intermediary components to reduce interaction latency, enforce security, and encapsulate legacy systems.

The idea of REST is that each SOA following REST constraints can be considered a compliant RESTful architecture and then completely interoperable though the web with each component well defined and thus fairly predictable. In a nutshell, REST constraints can be summarized in:

1. **Clien-Server.** Interactions between components are based on the request-response pattern, where a client sends a request to a server and gets back a response.

2. **Components Loosely Coupled.** Uniform interfaces implemented by all system's components are mandatory for achieving loose coupling between components.

3. **Stateless.** The client context and state should be kept only on the client, not on the server. Servers and applications can be stateful because this constraint simply requires that interactions between clients and servers contain information about each other's state.

4. **Caching.** Clients and intermediaries can store some data locally, which boosts their loading time, because that data does not need to be fetched from the actual server for each request.

5. **Layering.** Layered systems make it possible to use intermediary servers to further improve scalability and response times

Beyond reported constraints, REST is built upon 4 principles, briefly described as follows.

Principle 1 – Addressable Resources REST basically is a resource-oriented architecture (ROA), where every component of a system or an application is defined with the term *resource*. A resource can be individually addressed using an Uniform Resource Identifier (URI), also they can be organized in a hierarchy defined by a web path.

Principle 2 – Manipulation of resources through representations The tangible instance of a resource is called *representation*, which in REST must be a standard encoding of a resource using a MIME type.

Principle 3 – Self-descriptive messages REST emphasizes a uniform interface between components to reduce coupling between operations and their implementation. Every resource needs to support a common set of operations with clearly defined semantics and behaviour that can be identified in the standard HTTP methods GET, POST, PUT, DELETE and HEAD.

Principle 4 – Hypermedia as the Engine of Application State This fourth principle is centred on the notion of hypermedia, the idea of using links as connections between related ideas. REST requires the engine of application state to be hypermedia driven, as is each possible state of an application needs to be a RESTful resource with its own unique URI, where any client can retrieve a representation of the current state and also the possible transitions to other states.

3.2.2 Communication Protocols

From the network and communication perspective, IoT can be viewed as an aggregation of different networks, including mobile networks (3G, 4G, CDMA, etc.), WLANs, WSN, and Mobile ad-hoc Networks (MANET). Seamless connectivity is a key requirement for IoT. Network-communication speed, reliability, and connection durability will impact the overall IoT experience.

Unfortunately, building a single and global ecosystem of Things that communicate with each other seamlessly is a hard task to accomplish today. In fact, there is no unique and universal application protocol for the IoT that can work across the many networking interfaces currently available.

So, considering communication protocols, for the IoT an important issue is the exigence to deal with a single universal application layer protocol for devices and applications to talk to each other, regardless of how they are physically connected. An efficacious response to this exigence can be found in the Web of Thing approach – briefly describe in following section – where the Web become the universal communication protocol to be used for designing the IoT. See Figure 3.2 for a conceptual representation of this scenario.

3.3 The Web of Things (WoT)

The Web of Things (WoT) is a term used to describe a "supercharged IoT" firstly proposed by D. Guinard and V. Trifa in [33] as a way for designing the IoT with a unique vision in terms of application protocols, to provide interoperability to the IoT through the Web. In fact, rather than re-inventing completely new standards, the Web of Things reuses existing and well-known Web standards used in the programmable Web (e.g., REST, HTTP, JSON), Semantic Web (e.g., JSON-LD, Microdata, etc.), the real-time

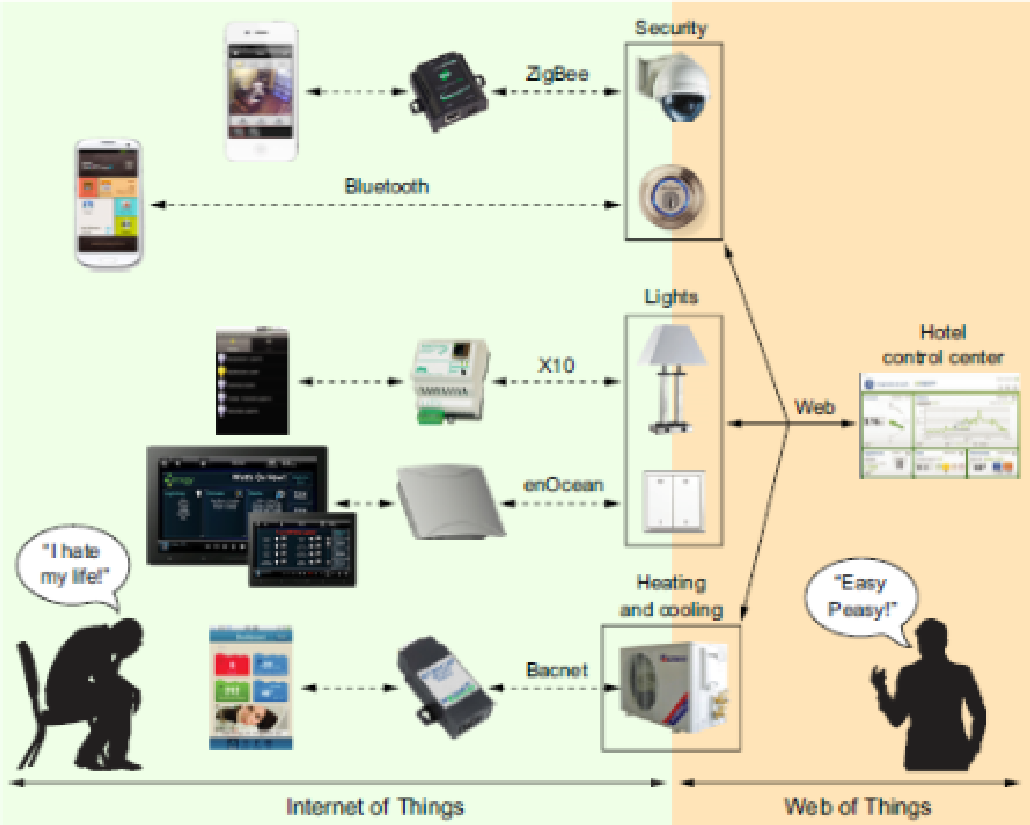


Figure 3.2: A pictorical motivation for introducing the Web of Things (WoT). This figure has been taken from [33].

Web (e.g., WebSockets) and the social Web (e.g., OAuth or social networks). In other words, by hiding the complexity and differences between various transport protocols used in the IoT, the Web of Things allows developers to focus on the logic of their applications without having to bother about how this or that protocol or device actually works.

Referring to IoT as described in the previous section, we can define better the WoT through the following considerations.

- **Easier to program.** Web protocols can easily be used to read and write data from/to devices, and are especially much simpler to use and faster to learn respect the complex IoT protocols. Besides, if all devices could offer a Web API, developers could use the same programming model to interact with any of them.

- **Open and extensible standards.** In the IoT, standards change very rapidly, sometimes are also not well documented and so is not simple to deal with them. Vice versa, the Web as the underlying layer in the WoT proposal ensures a stable starting point allowing to deal with standards that certainly will not change over time as quickly as other. Moreover, nowadays, HTTP and REST are an obvious choice when a designer wants to offer public access to some data.

- **Fast and easy to deploy, maintain, and integrate.** The maintenance of a pure IoT system is not an easy task to do, and often could be quite expensive in terms of the required amount of time and investments for infrastructures and tools. Vice versa, this task becomes very easy to do in a web-oriented vision, considering the nowadays degree of development of the web.

- **Loose coupling between elements.** HTTP is loosely coupled by design because the contract (API specification) between actors on the web is both simple and well defined. This allows any actor to change and evolve independently from each other. The ability for devices on the Internet of Things to talk to new devices as they get added without requiring any firmware updates is essential for a global Web of Things.

3.3.1 Architecture

Generally speaking, the Web of Things is a set of best practices that can be classified according to the Web of Things architecture. The architecture proposes four main layers (or stages) that are used as a framework to classify the different patterns and protocols involved. Unlike the ISO/OSI or the TCP/IP Stack, the WoT architecture stack is not composed of layers in the strict sense but rather of levels that add functionality, helping to integrate things to the web making them accessible for software applications and humans. The following paragraph will briefly describe those levels, while Figure 3.3 reports a representation of them.

Layer 1 – Access

This layer deals with the access of things to the Internet and ensures they expose their services via Web APIs. This is the core layer of the WoT as it ensures things have a Web accessible API, transforming them into programmable things. The access layer in the WoT is built around two core patterns:

1. all things should be exposing their services through a RESTful API;

2. integrates in the RESTful APIs a publish/subscribe mechanism to match the event driven nature of lot of IoT applications.

Some things can connect directly to the Internet but in other cases devices can access the Internet through Smart Gateways, that acts as protocol translation gateways at the edge of the network.

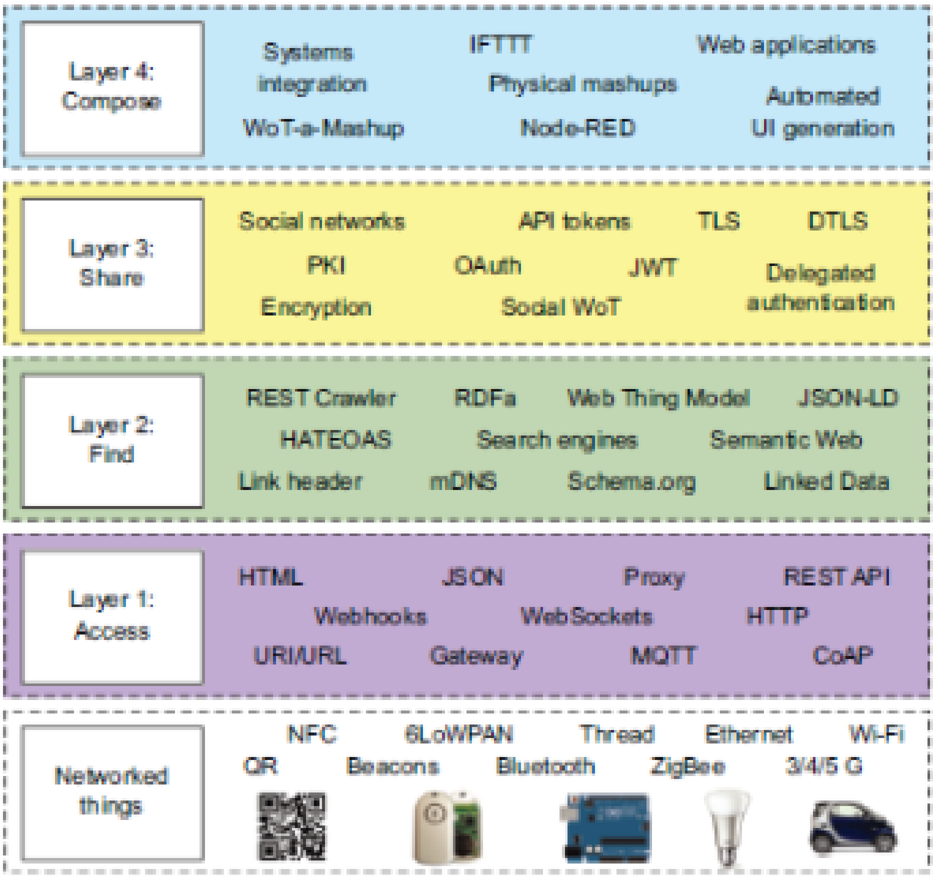


Figure 3.3: The stack architecture of the Web of Things. This figure has been taken from [33].

Layer 2 – Find

The goal of this layer is to provide a way to find and locate things on the Web and hence is strongly influenced by the semantic Web. The approach here is to reuse Web semantic's standards to describe things and their services.

Layer 3 – Share

The responsibility of the Share layer is about pushing data collected from thing to the web efficiently and securely. At this level, WoT applies fine-grained sharing mechanisms on top of RESTful APIs provided at Level 1.

Layer 4 – Compose

Finally, the goal of the Compose layer is to allow easily to create applications involving Things exploiting the web protocol. In other words, it is responsible for the integration of data and services from heterogeneous Things into an immense ecosystem of web tools such as analytics software and mashup platforms.

3.3.2 Real-time WoT

As reported in the previous section, at the basis of the WoT architecture there is REST as a way to represent things/resources over the Web offering an HTTP based protocol to access to them. With this protocol, clients always initiate the communication with a server by sending requests and expecting a response in return; this is known as request-response communication. Unfortunately, this protocol is not adequate when a more real-time interaction between clients and server is required, in particular when notifications need to be sent asynchronously by a thing to clients as soon as they are produced.

To achieve this goal, the Web of Things supports a publish/subscribe protocol for resources, allowing a further decoupling between data consumers (subscribers) and producers (publishers). Although other solutions were available, nowadays the community agrees on the fact that the publish/subscribe protocol for the WoT has to be implemented using HTML5 WebSocket support. In a nutshell, WebSockets enables a full-duplex communication channel over a single TCP connection. A WebSocket connection is initialized, creating a handshake in three steps. Firstly, a client sends an HTTP call to the server with a special header asking for the protocol to be upgraded to WebSockets. Then, if the web server supports WebSockets, it replies to the client acknowledging the opening of a full-duplex TCP socket. Finally, once the initial handshake takes place, the client and the server will be able to send messages back and forth over the open TCP connection.

WebSockets are very interesting to create areal time WoT. In fact, using the standard web technologies and opening a TCP connection over port 80, WebSockets are generally not blocked by firewalls and can easily traverse proxies. The permanent link created by WebSocket communication is interesting in an Internet of Things context, especially when considering applications wanting to observe the real-world properties such as environmental sensors but, of course, is not limited to this.

Chapter 4

Towards Pervasive Mixed Reality Systems

The brief analysis of available augmentation technologies showed how times are ripe to take into account novel kinds of software systems considering mixed reality and pervasive computing as their basic building blocks. During the last years, we have witnessed the birth of many software applications offering several degrees of mixed reality, in particular, games and advertising contexts. Generally, these applications have been developed as ad hoc systems, built from scratch considering available technologies and tools to meet well-defined needs and requirements. From the other side, pervasive computing is becoming a reality with many applications for the IoT, especially in industrial contexts and domotics. Anyway, at the best of our knowledge (see next sessions for a more in-depth analysis of state of the art), from the software engineering side, there is not available yet a shared model to guide developers that want to bring together Mixed Reality and Pervasive Computing in a simple but efficient way. In this dissertation we refer to this category of systems with the term *Pervasive Mixed Reality Systems*, considering those systems that integrate Mixed Reality in pervasive computing contexts to build wide, distributed, open and cooperative new smart environments to offer people new functionalities able to improve their ways to work and live.

4.1 Motivation and Objectives

Despite the scientific community (before) and the companies (later) have found in pervasive computing – in particular in the Internet of Things (IoT) – a concrete technological framework able to bring together the physical world and digital technologies, for a deeply and seamless conjunction of these two heterogeneous levels – reality and virtuality – there is still much work that can be done. In particular, from the research side, there are still many open questions to be investigated and possibly answered.

Although the IoT seems to be the accepted unique technological answer, a wide gap between the physical world – the reality where people live, work and interact – and the "digital world" – the software-based dimension – exists yet. The IoT vision is compelling to make the physical environment observable and accessible from the digital layer. With IoT, a software – and in the short future also an Artificial Intelligence (AI) – can observe and act upon those physical things that have been equipped with proper devices (sensors and actuators) to give them computational capabilities. IoT helps to create what we call smart environments – smart cities, smart buildings, smart vehicles and so on – but what it is not able to do is to bring humans in the loop.

Sure, using a smart device, a human can interact with the digital layer, can get real-time information by the physical environment, and is also able to exploits digital features to communicate with other humans and so on. By the way, this level of integration for humans in smart environments seems still "poor" and not very powerful as could be — the exploration of human augmentation technologies done at the beginning of this dissertation revealing us that there are many opportunities that we could consider to propose a deeper and more powerful integration of humans in smart environments. In other words, in the pure IoT vision, humans are not considered as first-class entities of the digital environment.

Considering what reported here and also what described in the previous chapters, emerge the opportunity to investigate how to fill better the gap between the physical and the digital, focusing on humans and environments to be suitably augmented by technology and in particular by the computation as a form of augmentation. For this reason, the central objective of this dissertation is to investigate how Mixed Reality can be put in synergy with Pervasive Computing to build new (really) smart environments where the and the physical should be strongly coupled, assuming both the digital layer as an extension of the physical one and vice versa. An integration where both humans and software agents [a] can interact and cooperate to improve ways of working and living in new environments we generally refer to with the term "*Augmented Worlds*".

Considering the given motivation that brings to envision the concept of augmented worlds, two are the main objectives of this dissertation. In particular:

1. Defining a conceptual model for Augmented Worlds, for the design of involved entities considering features imposed by the context of application (pervasive mixed reality systems). Doing this, the idea is to reach the right level of abstraction for the model, able to be a sound compromise between the generality of the approach and its applicability to the real world, considering available technologies and assuming interoperability as the main feature.

2. Designing and build a concrete software platform – based on the defined conceptual

[a] Here, with the term "agent" we refer to a generic proactive software component able to act autonomously considering a particular context.

model – to start prototyping this kind of systems with the purpose that platform becomes a reference platform for building new smart environments.

Besides these two main objectives, a third but less critical objective is to experiment several real case studies and applications for a concrete field assessment and evaluations of the proposed approach. This is important for a taste of what can be done with Augmented Worlds, avoiding to build in-lab case studies too simple for a reliable evaluation of open possibilities and perspectives.

4.2 Related Works

After describing the motivations behind the dissertation and the objectives that this arises, in this section a brief exploration of the state of the art about major related works is reported. What reported here constitutes a heterogeneous investigation into the literature because related works of what proposed in this dissertation come from several areas of interest. Some of them originate from the integration of distribution and cooperation features in mixed and augmented reality. Others, instead, originates from smart environments sector and artificial intelligence where augmented reality is introduced to enhance users' experience and to propose to them a concrete innovative computational support.

4.2.1 Outdoor Augmented Reality Gaming Systems

As often happens, several technologies leave research labs and become known for people because of their usage within captivating games. Partially, this is also the case of augmented reality that has started to become known in the mainstream at the beginning of the 2000s with a new version of the popular game *Quake*, called *Arquake*, re-proposed with the possibility to exploit an augmented reality experience [45]. In fact, using an old fashion head-mounted display (connected to a basic mobile computer and GPS system) to provide inputs to control the game, the user/gamer can walk around in the real world and play Quake against virtual monsters. The particularity that makes this game of interest for this dissertation comes from the fact that Arquake proposed a strong integration between the augmented world – the computer-generated information – and the physical reality. This game represents the first system that allows users to play augmented reality games outdoors with a first-person-perspective: the users can move around in the physical world and, at the same time, experience computer-generated (an managed) graphical elements. Also, a first attempt for a multi-user shared augmented reality experience was proposed by Arquake, exploiting an ad hoc local area network.

About fifteen years later than Arquake, other two augmented reality games have brought people's attention back on augmented reality. The first one is *Pokémon Go*, that in 2016 has been proposed as a location-aware augmented reality game, diffused to people through a smart-phone app. Moving around a city, the user has to detect and catch as much pokémons he can: the "catch" phase is proposed to users through a "poor" and not very seamless augmented reality experience. The second one, instead, is *KioskAR* that is a kind of pervasive augmented reality games making it possible for players to present their artworks in virtual kiosks placed in real-world locations, eventually shared with other players.

4.2.2 Context-Aware Mobile Augmented Reality

In literature, the concept of *Pervasive Augmented Reality* is used to define those software systems proposing a continuous augmented reality experience, providing always-on access to information integrated into everyday scenarios [32]. The primary requirement of this category of software systems is related to the ability to adapt to the current context of the user, as defined in Section 1.3. The challenge of those systems – that are user-centred systems – is to allow for each user to be involved seamlessly into an augmented reality experience in many contexts of daytime (at home, at work, outdoor during physical activities, etc.).

Due to the fact that context constitutes the main building block of those software systems, in literature has been defined the vision of *Context-Aware Mobile Augmented Reality* (CAMAR) as systems that focus only on the personalization aspects for mobile AR [41], possibly integrating other sources of informations not directly related to the user.

In CAMAR systems [60], especially in most recent studies about them, the following set of principles and features are recognisable.

- **Ubiquitous augmentation** – The real environment should be associated to a virtual one where users' perceptions and interactions are extended in combination with relevant information coming from the real environment.

- **High-level context-awareness** – Because there is a huge amount of information available from the environment and related to each user, it is important to understand each user current situation and allow to him/her to interact only with relevant virtual elements.

- **Sustainable participation** – Considering system where a lot of users want to join and participate, the infrastructure should be able to growth by need and re-adapt considering contextual information to collect.

- **Hybrid multi-object tracking and recognition** – Due to the fact that each real environment has it owns peculiarities, CAMAR systems needs to combine multiple techniques to recognize and interact with environments entities and elements.

- **Intelligent visualization** – While huge amount of contextual information is available, most of the mobile devices have a small screen size and only a limited sub-set of data should be proposed appropriately filtered considering user's needs.

4.2.3 Intelligent Virtual Environments

Although it refers mainly to the context of the digital virtual life, the literature on Intelligent Virtual Environments (IVEs) [37] is an interesting related work of what proposed in this dissertation, in particular, those contributions extending the basic IVEs towards the integration with the physical environment (e.g., [53, 54]).

In the research panorama, IVEs are the result of the combination between autonomous agents – equipped with intelligent techniques and tools provided by Artificial Intelligence (AI) – with significant graphical representation for both agents and digital environment [4]. In other words, they represent a convergence among several needs in artificial intelligence research, in particular, the need to incorporate aspects of the physical reality in environments where entities with intelligent behaviours live.

The primary objective of an IVE is to provide a context where simulate a physical/real environment with autonomous software entities as inhabitants, interacting with the simulated environment as they were the real entities (e.g., humans) of the simulated real world. These entities have to be presented to IVEs' users in an effective and useful way — e.g., with an appropriate graphical 3D representation. Moreover, the coherence between the simulation and the real world under simulation must be ensured and maintained.

4.2.4 Multi-User Collaborative Environments

In literature, a category of software systems enabling cooperation and collaboration among human users in virtual (and augmented) environments can also be clearly identified.

Croquet gives the first example in this area [62]. It is proposal for a system where involved elements have a distributed state which can change given the interaction with users and, more generally, given the computational activities and processes occurring inside the concrete application. In particular Croquet represented an attempt to proposing a computer software architecture mainly oriented on deep collaboration between teams of users. In this case, except for human users, elements/objects composing the mirror worlds are only virtual and always deployed in the digital layer.

A Croquet similar approach on multi-user collaborative environments is proposed in [12] introducing the Manufaktur prototype. This project represents an attempt to go beyond traditional computer-based virtual environments towards a collaborative augmented shared environment where users can be immersed in a kind of augmented desktop where documents and objects appear as live representations in a 3D workspace.

Finally, another example of a system where real environments are reflected into cyberspaces to augment collaboration opportunities for humans is reported in [35]. Here, the primary purpose of the digital layer is *to acts as a sort of community-level perceptual system*, to support the activities and awareness of the involved people.

Collaborative Mixed and Augmented Reality

A special attention in exploring literature about collaborative augmented environments must be devoted to analyse some works proposed by M. Billinghurst and H. Kato – that are two well known and estimated members of MR/AR community. In particular we refer to studies on *Collaborative Mixed Reality* [6], published in 1999, and *Collaborative Augmented Reality* [7], published in 2002.

In the first contribution on Collaborative Mixed Reality, authors consider mixed reality techniques for developing Computer Supported Cooperative Work (CSCW) interfaces. They identify two main motivations in developing such interfaces using MR. In particular the possibility (1) to extend classical CSCW with a seamless integration in the real environment and (2) to go "beyond being there" toward a collaboration not only face-to-face.

The second contribution can be considered as an extension of the previous. In this work the vision of an augmented CSCW for users is extended to the possibility to enhance offered functionalities with virtual object that can be manipulated by users and shared among them.

Independently of which contribution we consider, emerge clearly from those work that the integration of Mixed Reality with the real environment could be clearly change and improve the way in which people live and work in.

4.2.5 Intelligent Virtual Agents

Intelligent Virtual Agents (IVAs) can be defined as embodied digital characters situated in a mixed reality environment that look, communicate, and act like real creatures. A primary feature of IVAs is that their behaviour should exhibit some aspects of human intelligence – i.e., autonomous behaviour, communication and coordination with other IVAs and, potentially, learning capabilities. Generally, these Virtual Agents are expected to be believable, i.e., to have consistent behaviour, to exhibit some form of personality and emotions, to communicate and interact plausibly. Examples of IVAs are: non-player characters in computer games, virtual tutors in educational software, virtual assistants or chat-bots, virtual guides in digital heritage environments, and so on.

The design and development of virtual agents require the adoption of theories, models and tools from related disciplines, including Artificial Intelligence, Artificial Life, Mixed Reality and Human-Computer Interaction.

Intelligent virtual agents software has improved with advances in AI and cognitive computing. Current conversational technology allows virtual agents to go far beyond interactive voice response (IVR) systems; virtual agents understand users intent and can provide personalised answers to customer questions in a human-like manner. An avatar is often used to provide a visual representation of the virtual agent.

Generic issues, such as design and evaluation methodologies, development tools and platforms, and believability assessment, as well as more specific issues related to agent behaviour, interaction and communication, are under active research. Furthermore, the recent emergence of novel interaction techniques and devices, including mobile augmented reality, natural user interfaces and low-cost immersive VR, presents further challenges for IVA embodiment and communication with humans.

4.2.6 Mixed and Augmented Reality Cognitive Agents

Other related works of particular relevance for this dissertation have been introduced and studied within the area of Multi-Agents Systems (MAS) [66, 27, 69], especially in the context of Cognitive Agents based on the Belief-Desire-Intention (BDI) model [47, 31].

In literature, a comprehensive description about the role of agents and MAS in Mixed and Augmented Reality contexts has been given with the concept of MiRA (Mixed Reality Agents) [34] and AuRAs (Augmented Reality Agents) [13]. The same group of main authors proposes both works. The focus of the investigation of MiRA and AuRAs is about agents having a representation in an AR/MR environment making them perceivable by human users and enabling interaction with them, as well as with other agents.

Avoiding entering in details – even though both works referring to similar concepts – we can consider MiRA as the broadest concept, used to delineate "an agent living in a mixed reality context", blending the notion of agency with embodiment [22] and Milgram's reality–virtuality continuum. Vice versa, AuRAs identify those mixed reality agents that can both sense and act in the virtual component of their reality but can only sense in the physical. This specialisation helps distinguish AuRAs from MiRA with a primarily physical component, such as robots operating in an augmented reality environment.

MiRA – Mixed Reality Agents

The aim of studies on MiRA is to offer a comprehensive survey of state of the art considering convergences between mixed reality and cognitive agents. Defining a mixed reality agent as an *agent embodied in a Mixed Reality environment*, this work proposes and reflects on a taxonomy to categorize those agents considering following features:

1. **Weak vs. Strong Agents** – A week agent [70] is an autonomous entity, (*i*) able to interact with other agents, (*ii*) able to perceive from the environment reacting/acting over it and, finally, (*iii*) able to taking initiatives proactively considering its

goals. A strong agent [61], instead, is a weak agent with the addition of mentalistic attitudes like beliefs, knowledge and obligations.

2. **Virtual vs. Physical Corporeal Presence** – Corporeal presence is more than simply visual representation. Different degrees of features such as occlusion, collision detection, or adherence to other physical laws, increase the corporeal presence of virtual entities. For physical entities, instead, the corporeal presence assumes a more restrictive significance — e.g., a physical entity cannot escape from physical laws, and so on.

3. **Virtual vs. Physical Interactive Capacity** – The interactive capacity represents the ability of the agent to sense and act both in the virtual and physical layers. Considering nature of different agents, this capacity could range from the interaction in pure virtual style (e.g., for those agents with a completely digital corporeal representation) to the full physical interaction (e.g., for those agents embodied in robots).

Results of this categorization converge into a MiRA Cube where above features for mixed reality agents represent the three-dimensional axis. This taxonomy is, nowadays, strongly accepted and can be used to classify a wide category of works in this research filed, as been done by authors.

AuRAs – Augmented Reality Agents

As reported above, the concept of AuRAs is different from the concept of MiRA, because – according to authors – an augmented reality agent can be categorized in the MiRA cube as a mixed reality agent with the ability to sense both in the digital and in the physical context but able to act only in the digital one. In this meaning AuRAs excludes all agents embodied in physical (autonomous) devices such robot for example.

Moreover, the authors identify two main purposes in developing augmented reality applications involving cognitive agents:

1. **AuRAs as interface paradigm**, modelling interface agents living in augmented reality environments and directly interacting with human users.

2. **AuRAs as design paradigm**, considering a direction for an effective software development solution tackling main complexities of the real-world environments characterising augmented reality applications.

Finally, beyond identifying in AuRAs a concept for the development of augmented reality applications – according to authors – AuRAs could also be considered as a *powerful metaphor* for designing and building an “*aura*” around a human user as a form of augmentation in terms of features and capabilities improving the interaction of humans with digital layers.

4.2.7 Mirror Worlds

The term "*Mirror World*" has been introduced in the early 90s of the twentieth century by D. Gelernter to define *software models of some chunk of reality*, that is *some pieces of the real world going on outside the window endlessly fed by oceans of information through software pipes* [30].

In Gelernter's vision – using his words – a Mirror World represents *a true-to-life mirror image trapped inside a computer*, which can be then viewed, zoomed, analysed by real-world inhabitants with the help of proper software (autonomous) assistant agents. The primary objective of a mirror world is to strongly impact on the life of the citizens of the real world, offering to them the possibility to exploit software tools and functionalities provided by the mirror world, generically, to tackle the increasing "life complexity".

From a modelling point of view, in the original idea of mirror worlds, tuple spaces [29] represents the coordination media where information from the physical world are stored and then queried by software agents using Linda's coordination primitives. In other words, tuple spaces are used as a way to augment the environment to organise information feeding the digital layer.

Mirror Worlds as open societies of cognitive agents

Inspired by the Gelernter's vision, more recently, a concrete research work on mirror worlds has been carried on in the context of agents and multi-agents systems [49]. Surely, this work is the primary source of inspiration for the idea of bidirectional augmentation within the "physical–digital" loop adopted in this dissertation.

In this work, a mirror world is conceived as an open society/organisation of cognitive software agents spatially layered upon some physical environment, augmenting its functionalities [50]. Mirroring occurs when physical things, which can be perceived and acted upon by humans in the physical world, have a digital counterpart (or extension) in the mirror world so that they can be observed and acted upon by agents. Vice versa, an entity in the mirror world that can be perceived and acted upon by software agents can have a physical appearance (or extension) in the physical world – for example, through augmented reality – so that it can be observed and acted upon by humans.

This research work on mirror worlds offers an interesting framework where investigate the integration of different technologies for the definition of open computer-supported collaborative environments where human and software agents implicitly interact and cooperate. From a philosophical perspective, authors think the coupling between the physical world and the corresponding mirror digital layer could have a deep impact on humans and their sociality. In particular, they reason about consequences like *Social*, *Cognitive* and *Temporal Augmentation*. Instead, from a technical perspective, they also offer a basic practical framework based on the JaCaMo platform [8] to design and simulate mirror worlds. In particular, the CArtAgO framework [51] has been extended to support situated workspaces and situated artifacts as extensions of regular ones.

Related research in the field of mirror worlds that considers cognitive agents is proposed in [24], where a mirror world is built to allow software agents to increase their knowledge for general reasoning about the real world. Furthermore, users are allowed to use the mirror world, with its visualised data and highlighting of the agent's reasoning, for further understanding of the agent's behaviour, for debugging, testing or simulation purposes.

4.3 Contribution of the Dissertation

The exploration of the state of the art and in particular the literature on Mirror Worlds offers an interesting starting point to introduce the contribution behind this dissertation. Our work is strongly related to mirror worlds—which can be considered the "root" of the idea of the augmented world, especially if we take into account the initial Gelernter's proposal. Mirror Worlds propose a broader conceptual framework than augmented worlds for envisioning future smart environments. In particular, Mirror Worlds are not limited to hybrid environments where augmented reality is the medium to perform interactions between the physical and the digital layers.

On the one hand, the augmented world vision can be considered an extension of the mirror world model (see next sessions for more details), expanding it towards a concrete adoption in the design and engineering of real-world (agent-based) pervasive mixed reality systems. On the other hand, instead, mirror worlds go further than pure pervasive mixed reality systems. A mirror world can be, e.g., uses to simulate a real context, or to analyse behaviours in distributed environments even in the case in which human users are not involved.

Regardless of whether they could be considered as an extension or a subset of the mirror worlds vision, augmented worlds want to propose a concrete framework of reference for modelling, designing and developing of future smart environments. In this direction, focusing on pervasive mixed reality based smart environments, augmented worlds introduce novel concepts and requirements that are not contemplated by mirror worlds or, there, can be considered more lightweight that in augmented worlds.

4.3.1 Recognizable Benefits and Relevance

The vision of Augmented Worlds (better explained and discussed later in this dissertation) is about proposing a model and a platform to design and build Pervasive Mixed Reality Systems where the distinction between the digital and physical world should be as much as seamless. In a nutshell, augmented worlds are distributed and multi-users software systems where Mixed Reality acts as a glue to integrate into the physical world a pervasive computational layer with proactive digital entities coupled in a specific location of the physical layer.

Augmented Worlds are based on a notion of "augmentation" which is wider than the sum of MR one: by also integrating the pervasive computing perspective we deal with an augmentation that is about enriching the physical environment with computational capabilities exploitable in terms of mixed reality by involved humans and teams of humans.

Among others, the main benefits emerging from the Augmented Worlds proposed approach are:

- a conceptual model to engineering with a high level of abstraction, novel kinds of software systems able to exploit augmentation technologies both for humans and for the environment;
- an integrated solution for exploiting Mixed Reality techniques in the context of IoT avoiding ad hoc implementations preferring a general-purpose direction;
- a contribution to the state of the art, with an alternative solution to deal with requirements of next future smart environments.

Next sections will be devoted entering into the details of the augmented world vision, exploring the involved concept for the formalisation of a conceptual meta-model. Nevertheless, considering the analysed state of the art, the work proposed in this dissertation well fits in the context of the involved research panorama, filling a conceptual and practical gap that does not seem to have been sufficiently delved into, at least in the direction followed by this dissertation.

Part II

Augmented Worlds

Chapter 5

Vision, Concepts and Model

In the previous chapter, we have introduced the term *Augmented Worlds* (AWs) to gather in a single category those software systems that integrates pervasive computing with mixed reality to shape hybrid smart environments where the physical world – involving both things and humans – is strongly coupled with one or more digital/computational layers. This emergent category of innovative and challenging software systems represents the vision and the objective of this dissertation. In particular, a proposal for the design and the engineering of this kind of systems is discussed here. considering the technological background detailed in previous chapters, here we describe our research about how to conceive this new category of smart environments we generally call Augmented Worlds. We will describe and analyse their features, main concepts and challenging aspects, toward the definition of a complete and effective conceptual meta-model for their design. This meta-model aims to capture in a uniform way the main concepts of an AW, to offer designers the appropriate abstractions to model augmented world's specific instances.

5.1 What is an Augmented World?

Generally speaking, the concept of Augmented World (AW) captures in a single uniform model main aspects of augmented technologies and their integration/convergence, to design and develop smart environments as collections of proactive autonomous software components interacting with real things, physical space and its inhabitants.

In AWs the physical space becomes a first-order concept of the computational layer, beyond the perspectives already developed in distributed and mobile computing toward a vision of full pervasive computing (as previously described in Chapter 3). In literature, a related vision about considering the physical space as a fundamental concept has been developed so far in Spatial Computing [72, 59], even if with a perspective that is quite different from the AWs' one. Spatial computing systems are typically based on a very large set of mobile computing devices locally connected and location-aware that

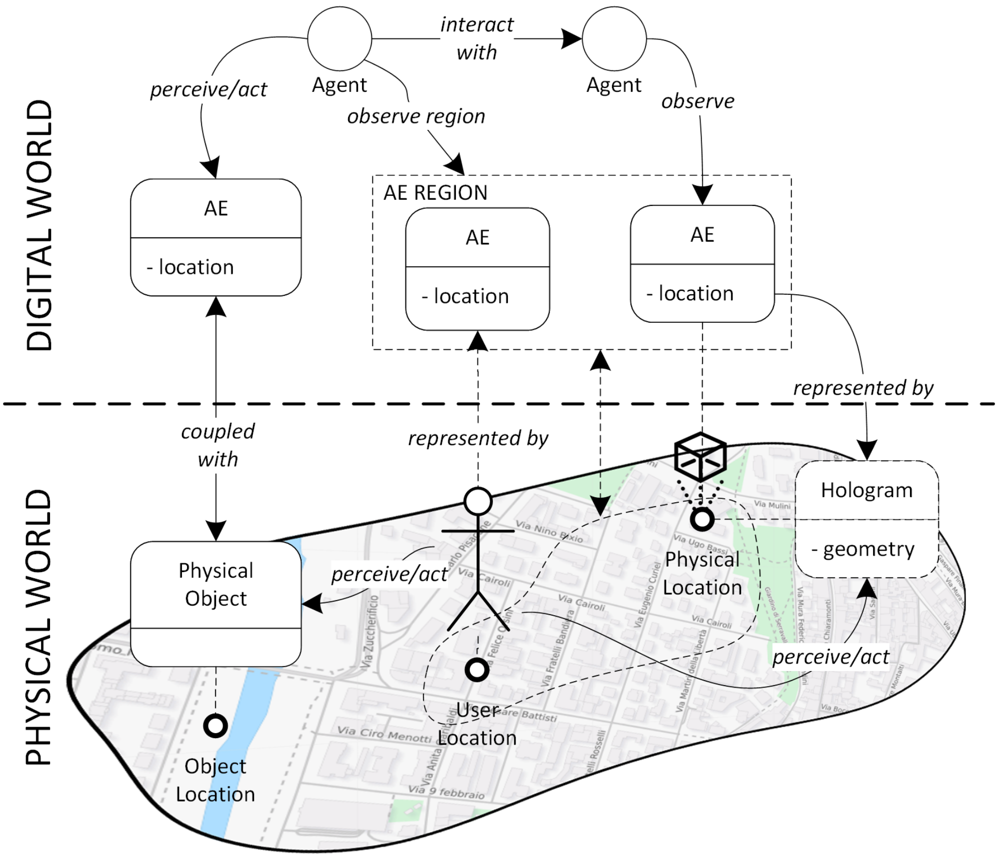


Figure 5.1: A conceptual representation of an Augmented World.

support the execution of distributed computations in which the spatial configuration of the elements has a primary role. In AWs, instead, the concept of space is mostly a logical concept: both physical and computational entities are featured by a notion of position within the physical space but – for the computational ones – is not required that the computation were executed in a computational device placed in that position.

5.1.1 Definition

An Augmented World can be intuitively defined as:

> *a software application that enriches the functionalities of a particular physical environment (e.g., a room, a building, a city, etc.) by means of full-fledged computational objects located in a particular location of the physical space, that both users and software agents can perceive and interact with.*

Referring to these computational objects as *augmented entities* (AEs), we can imagine (a portion of) the physical world where humans live and work, coupled with one (or more) digital layer(s) augmenting the real world and his inhabitants with computational capabilities exploitable both by humans and thing of the environment. Figure 5.1 provides a first taste of what we mean (more details about involved elements will be discussed in Section 5.3 where the AWs conceptual model will be introduced).

In particular, an Augmented World is a *hybrid environment* where both humans and proactive autonomous entities can interact and cooperate through the digital layer that offers a way to spread computational entities/agents into the physical environment at specific locations augmenting – in terms of functionalities and technological-based opportunities – the environment itself and its inhabitants. In an augmented world, according to the well-known augmented reality definition, each user has its perspective over a shared world and can interact with other users and agents, observing and acting upon augmented entities. The physical world is taken in account, first of all, considering smart things as in the IoT vision, but in AWs' vision we are not limited to this: either non-computational physical things can be involved in an AW somehow.

5.1.2 Coarse-grained Features

Before entering in details of the conceptual meta-model (see Section 5.3), from a high level of abstraction an Augmented World can be described as a complex and smart software system exposing following coarse-grained main features.

Multi-user context and cooperation. First of all, considering the definition given in previous sections, an augmented world is characterised by a context where multiple human users could be concurrently involved. Each user, equipped with a proper computational device, must be allowed by the system to enter in the mixed reality environment mainly for both observe and interact with digital entities. The system has to perceive the presence of each user, allowing to all of them to cooperate and collaborate exploiting provided functionalities. Also, users can also be managed as teams, providing to each team specific extra features.

Interactive context. Concentrating on the single user, him/her interactions within an augmented world can be "passive/reactive" or "active". In the first case, the human user can be considered just as an observer of the augmented world, perceiving elements but without the possibility to interact directly with digital entities or other users. In this case, a user can be eventually notified by the augmented world infrastructure of some occurring events. In the second case, instead, the user has the full control of the augmented world with the ability to interact with digital entities. For example, if a digital entity is represented in the augmented world as a hologram, the user could want to move/touch it, going beyond the simple and commonly accepted augmented reality oriented interaction mainly based on the visualization and observation of the computer-generated images. As another example, a user can interact with a physical thing, and this interaction should be perceived and registered in the augmented world to eventually propagate an event bout this to the involved actors. Vice versa, considering the set of users involved in the augmented world in a given instant of time, the interaction among them have to consider their actions within the augmented world proposing a uniform way to improve direct and indirect cooperation/collaboration.

Shared and distributed Environment. Considering the intrinsic nature of an augmented world that, is obvious to suppose that each instance of such a system should be considered (and deployed) as a distributed system. Anyway, from a logical point of view, each instance of an augmented world must be considered from involved users and agents as a concentrated system. Moreover, this logical instance must be the same for all involved actors: in other words, the instance of an augmented world and its digital entities must be shared among users and agents.

Real-time execution. The previous requirement about having a shared instance of an augmented world for users and agents and the physical world strong coupling imply that an augmented world has to be executed considering the real-time constraint. This feature is important at any level. From the users perspective, since the current state of the world's entities is shared by all, each update to this state performed by a user should be instantly available to other users – taking in account concurrency issues coming from concurrency actions performed by different users on the same entity. Also from agents perspective, the constraint of real-time is mandatory if these entities need to reason on and act upon the current state of the digital world. Finally, because the digital layer is strongly coupled with the physical world (see next proposed feature for details), the digital state must be rapidly informed and updated according to real-time changes of the physical world.

Bidirectional augmentation. In an Augmented World, the digital layer can be considered as an extension of the physical one, but also the physical layer can be considered an extension of the digital one. This represents an essential feature for AWs and means that an augmented world can have two main point-of-view. Firstly, the most obvious, a software system should be conceived as an AW when there is the need to enrich a particular physical environment (e.g. a building, an office or an industry) with pervasive computational capabilities. Secondly, but not least, AWs can also be used to introduce physical elements and humans in a completely digital environment (e.g., a business intelligence process with the exigence to reason about humans actions and behaviours). Furthermore, the concept of *bidirectional augmentation* can also be intended as a closed loop between physical and digital layers. The physical world is augmented by the entities living in the digital layer (e.g. in the augmented reality meaning or considering the availability of smart things able to both communicate their current state and receive commands from a software entity). Vice versa, the digital layer is augmented by data and information received by humans (e.g. their current position, gaze and interactions) and physical things (e.g. their shape, position, and so on).

Dynamism. In an augmented world both human users and software agents can dynamically join and leave. Moreover, also the structure of each AW – regarding the set of virtual entities/holograms – can dynamically change due to agents/humans (direct or indirect) actions at runtime.

5.1.3 Challenges

The vision on AWs exposes some interesting aspects – some challenges – that will be considered in the conceptualization of the meta-model. Other challenges that will arise from the engineering of a framework supporting the design of augmented worlds will be considered later. Starting from the given definition and focusing on the idea of using located augmented entities as a medium to allow the interaction between the real world and digital one, clearly emerges how the Augmented Worlds approach has the opportunity to open new research perspectives on how to design and build such pervasive mixed reality systems homogeneously, going beyond ad hoc solutions and implementations. New perspectives that will be only reached dealing with exposed challenges and finding an appropriate solution for each of them.

Beyond the "standard" Augmented Reality. Augmented Reality offers a way for adding computer graphics generated objects to a user's view on a portion of the real world. Thanks to some successful examples, people have started to experience some degrees of what these new techniques can offer. Anyway, the availability to place virtual object "into people's reality" represents only a portion of what could be developed and proposed exploiting emerging technologies. In fact, a guideline for the research carried on and described in this dissertation is that in the design of new pervasive smart environments the dare is not only related to develop some specific (software) applications with an effective tuning in the phase of registering digital elements with the real world – e.g., to make sure that if a physical object hides a portion of a digital one, the latter is only partially shown – also ensuring an effective coupling between the two layers. From the mostly passive role of computer-generated elements proposed by standard augmented reality approaches, nowadays we are assisting to a scenario where digital entities become increasingly more (pro)active, especially in interacting with humans and physical environment. Just as an example, digital entities can detect and respond to users' gestures and movements. They can react to users gaze or indirect interactions. Moreover, the digital environment can be instrumented to observe and understand the physical counterpart, e.g., to react to changes in non-computational objects of the real world. As reported in Chapter 1, we can assume both humans and environment augmentation.

Building an overall belief in the Augmented World current state. A challenge of the Augmented Worlds approach can be found within its definition: allow to autonomous entities and humans to *easily* join to an augmented world and interact each others considering the stream of data/information flowing from physical to digital layer and, vice versa, used to build the overall real-time belief of the augmented world inhabitants. From an infrastructural perspective, an AW is a shared and multi-user distributed systems: despite this, two or more users must perceive the same observable state of the shared augmented entities even if they enter in the AW's running instance with different computational capabilities and with heterogeneous devices and tools.

A "degree of freshness" for data and information. Considering the level of real-time coupling and synchronization between the computational augmented layer and the physical one, another significant challenge is given by the fact that the coupling/synchronization is critical from users' perspective, since it impacts on what users perceive of an augmented world, and then how they reason about it and act consequentially. If a part of the augmented world is temporarily disconnected – because of, e.g., some network transient failure – users must be able to realise this. These aspects are challenging in particular because – like in distributed systems in general – it is not feasible in an augmented world to assume a single clock defining a centralised notion of time. Conversely, it is fair to assume that each augmented entity has its local clock and the events generated inside an augmented world can be only partially ordered.

Causal Consistency. Despite the distribution, causal consistency must be guaranteed, in particular, related to chains of events that span from the physical to the digital layer and vice versa. That is, if an augmented entity produces a sequence of two events concerning the change of its observable state, the same sequence must be observed by different human users immersed in the augmented world.

Unpredictability. Also, the spatial coupling and physical embedding properties of augmented worlds introduce further elements and complexities, that are not fully captured by strategies adopted in software smart environments where a complete virtual approach is adopted. Taking into account the real-time, dynamic, heterogeneous and possibly unpredictable physical environment is the primary challenge of the augmented world vision. When humans are involved in the software as first-class entities and not as just external users, critical issues grow a lot.

Ethical Issues. Augmented Worlds are hybrid environments where an open society of intelligent agents lives, layered on top of physical environments where people live, providing different levels of high-level augmentations, from cognitive augmentation to social augmentation. This view triggers a wide range of research issues and challenges that have not been explored yet about this kind of augmented, blended societies in terms of models (including organizational models), infrastructures, as well as the problems and risks that such a vision about augmented societies can bring, and consequently devising the proper ways to avoid or deal with them.

5.2 Multi-Agents Systems in the loop

So far in this dissertation, the term *agent* has been used to generically refer to those software entities exposing a certain degree of autonomy both in the observation of the current state of an AW and in the interaction with the other entities involved. In the research literature, such vision well fits in Multi-Agents Systems (MAS) studies and, in particular, to those agents defined as Cognitive Agents.

Before entering in details – in particular, for a brief introduction to MAS and to discuss the role that agents can have in the design of Augmented Worlds – we can state that agents can play a key role in Mixed Reality first of all for modelling and implementing holograms that need to feature an autonomous behaviour, eventually interacting with other holograms[a] and with humans, as well as with the physical world where they are immersed.

More generally, we can consider augmented worlds as agent-based pervasive mixed reality systems where agents can dynamically create and control virtual objects and holograms, along with the control of physical things, to devise new kind of effective smart environments. Besides individual holograms, multi-agent systems and agent organisations can be fruitfully exploited to model complex and possibly open, large-scale systems of holograms, featuring some degree of coordinated, cooperative, social behaviour.

5.2.1 A brief introduction to Agents and MAS

Multi-Agent Systems (MASs) are software systems composed of multiple interacting computing entities known as *agents*. An agent is a computational entity that is situated in some environment, and that is capable of autonomous actions in this environment in order to meet its design objectives [70]. Main recognizable features of an agent are:

- **Autonomy**, agents operate without the direct intervention of humans and have some kind of control over their actions and internal state;

[a] At this point, the term *hologram* is used to indicate a virtual geometry in a Mixed Reality environment.

- **Situatedness**, agents are entities situated in some environment, and they are able of sensing their environment (exploiting sensors) and of acting over it (through actuators);

- **Pro-activeness**, agents can decide for themselves what they need to do in order to satisfy their design objectives;

- **Reactivity**, agents can react to both events coming from the environment and communications received from other agents;

- **Interaction**, agents interact with other agents (and possibly humans) via some agent-communication language.

Historically, Multi-Agent systems have been studied since about 1980, but only several years later (in the 1993) Y. Shoham [61] coined the idea of Agent-Oriented Programming (AOP) as an innovative programming paradigm combining the use of notions generally referred to human's brain (such as the ability to believe that something is true or false in a specific moment of the life with a certain degree of certainty) for programming set of autonomous agents in a *societal view of computation*. Nowadays, multi-agent systems are generally used to develop complex distributed smart systems, where not only entities are present, but they need to interact in complex ways and need to have social structures and norms to regulate the overall social behaviour that is expected of them.

Cognitive Agents

Over the years, several agent models and architectures have been proposed. Among these, the Belief-Desire-Intention (BDI) agent model [47] is nowadays becoming the reference one used in state of the art for design and building cognitive agents within MAS. In the BDI model, an agent is featured by a mental state which is defined by *beliefs*, *desires* and *intentions*, described as follows:

- **Beliefs**, the information that the agent has about the world and other agents part of the MAS. The belief represents the context that agent has about the entire system and its involved entities;

- **Desires**, represent objectives or situations that the agent would like to accomplish or bring about. In particular, in the BDI dictionary, the concept of *goal* refers to a desire that has been adopted for active pursuit by the agent;

- **Intentions**, represent the deliberative state of the agent, in other words, what the agent has chosen to do. In MAS, this means the agent has begun executing a *plan*, i.e. a sequence of actions that an agent can perform to achieve one or more of its intentions.

A cognitive BDI agent acts accordingly M. E. Bratman *practical reasoning* [10], a decisional process that brings each agent from intentions to actions considering the belief. In particular, practical reasoning is characterized by two main phases: (1) the *deliberation* phase, when the agent decides which are the intentions it wants to achieve and (2) the *action* phase when the agent decides how to achieve a chosen intention using the available plans and actions. Concluding, a BDI agent can be considered like a *reactive planning system*, whose computational behaviour follows the practical reasoning, on the basis of the agent control loop[b].

The Environment Dimension for Cognitive Agents

The notion of environment is a fundamental concept in MAS, representing the virtual or physical context where agents are situated, which agents are capable of sensing through some sensor, and modifying executing actions provided by a set of physical or digital actuators. In the recent, the concept of environment has been investigated in [68]. In this work, authors assert that *the environment is no longer just the target of agent actions and the container and generator of agent percepts, but a part of the MAS that can be suitably designed to improve the overall development of the system.* So, the environment should be conceived as something that can be designed to be the right place for agents to live and work within.

With this notion of *environment* in mind and considering the given definition of augmented worlds, it is clear how the concept of environment become more and more important if we mix MASs with humans and (computational) physical things, towards pervasive mixed reality systems with a strong presence of cognitive agents. In literature, several models for designing agents environments have been proposed. Some of them will be taken into account for defining augmented worlds conceptual model (in particular the A&A model [42, 52], briefly described in the introductory part of Section 5.3). Nevertheless, from agents perspective, Augmented World can be considered as another way to designing environments. A way that takes into account not only the digital layer but brings in the loop also the physical one with humans and physical things as first-class entities.

5.2.2 Agents and MAS for Augmented Worlds

As previously discussed, the reference context of an AW is a physical environment – inhabited by human users engaged in some physical (individual or cooperative) activities, that could require mobility, physical actions, etc. – coupled with an augmentation layer that aims at enriching/extending the real one with a set of functionalities mainly but not only designed for supporting/helping human actions. In this context, cognitive agents

[b]To deepener how the control loop of an agent works, that goes outside the scope of this dissertation, the reader is referred to [9].

can be introduced in AWs first of all to developed humans counterparts in the digital world, that is: the proactive entities able to reason and act within the augmented world both exploiting augmented entities and cooperating/collaborating with human users.

Despite this, the role that agents and MAS can play in the area of Augmented Worlds is broader and also offers new interesting opportunities for the research context. We can distinguish among two different levels, following described.

A new era of Software Personal Assistant Agents

If we consider the information provided by the AWs context about humans actions and real-time perspective, agents augmenting the users allow conceiving a new generation of software personal assistant agents (SPAs).

Existing proposal and technologies have been developed for different kind of purposes and capabilities – from scheduling joint activities, monitoring and reminding users of key timepoints, sharing key information, assisting in negotiation decision support. Generally speaking, with SPAs the user is engaged in a cooperative process in which human and computer agents both initiate communication, monitor events, and perform tasks. This is based on the metaphor that of a personal assistant who is collaborating with the user in the same work environment. Augmented Worlds introduce a richer context that allows for further interesting functionalities to explore.

The main attractive feature of SPA agents in AW is the capability to "*see what the user sees*" – e.g. because in the AW the gaze of a human user can be represented and tracked augmented entities – and, more generally, to know what the user perceives about current context. An agent is capable of observing and initiating user interaction with the environment (which is meant here also to include perceptions). This allows to frame new kind of pro-active assistance in which SPAs reason not only about the context of the user but about what the user is perceiving (and not perceiving). These agents are meant to build dynamically a model about what the user is perceiving, and use this knowledge along with the information about user's goals, the state of ongoing activity, the actual state of the physical environment, to provide a proactive and smart assistance, possibly anticipating and notifying problems and suggesting actions to do. This scenario can be further enhanced by considering SPAs assisting teams of users immersed in the same augmented world, possibly sharing the same physical context or being in different places. In that case, different SPAs can communicate and share information (through the network) in order to understand what the team sees overall and use this knowledge to help both the action of the individual user and the coordination of the team.

Agents as Mediators for Augmented Realities

So far we have considered users and their SPAs immersed in purely physical environments: further interesting issues arise as soon as we introduce augmented reality and, more generally, augmented environments. In this case, the kind of assistance of SPA can also include the possibility to interact with/manipulate the augmented entities enriching the physical space, possibly shared with other users (and SPAs) situated in the same space. For instance, a SPA could automatically annotate the environment with virtual notes on the physical environment where its user is working – e.g., on physical objects used by the users – as a memo for the user herself or useful for implicitly coordinating with other users, perceiving those virtual signs. Moreover, differently from a generic mobile augmented reality browser, a SPA could show only those augmented entities that the agent knows to be relevant according to users' goals and needs for current activity, so functioning a smart filter when moving in complex AW.

These scenarios call for investigating the role of mediators and indirect interfaces for SPAs in AW, and the kind of delegation that can be assigned to them.

5.3 The Conceptual Model

The given definition for Augmented Worlds – described in previous sections together with the list of coarse-grained functionalities required from AW – allows to envision what an AW is, but it is not self-explanatory about which the main components composing an AW are and about their design features. To better clarify the given definition and to concretely define first-class entities composing AWs, this section introduces and describes a conceptual model that can be considered the reference guide in terms of concepts and relationships among them to be considered for the design of AWs.

Honestly, it would be more appropriate to refer to this model as a "meta-model". In fact, it describes the abstractions to be used to modelling the application logic of each instance of AWs from a design and programming point-of-view. Anyway, in this dissertation we will refer simply to it as conceptual model, exploiting the double feature of being both a kind of formalization for AWs and, also, the starting point for designing and building a concrete software platform to support the development of AWs (see next part of this dissertation for more details).

Considering the nature of augmented worlds as pervasive mixed reality software systems with a strong agent-based part (from now on the term agent will refer to the concept of cognitive agent according to the definition given in Section 5.2), it is worth clarifying that the proposed conceptual model is inspired to a well-known category of models and paradigms proposed within the MAS literature for modelling the interaction between software agents and the environment (either physical or virtual). In particular, as previously mentioned, the conceptual model proposed in this dissertation has been strongly inspired by the A&A meta-model, that is a model for designing and engineering MAS that introduces concepts of *artifact* and *workspace*, along with agents, as first-class abstractions [42, 52]. Briefly, in the A&A model, a MAS is conceived in terms of a set of agents that act together within a common working environment exploiting different

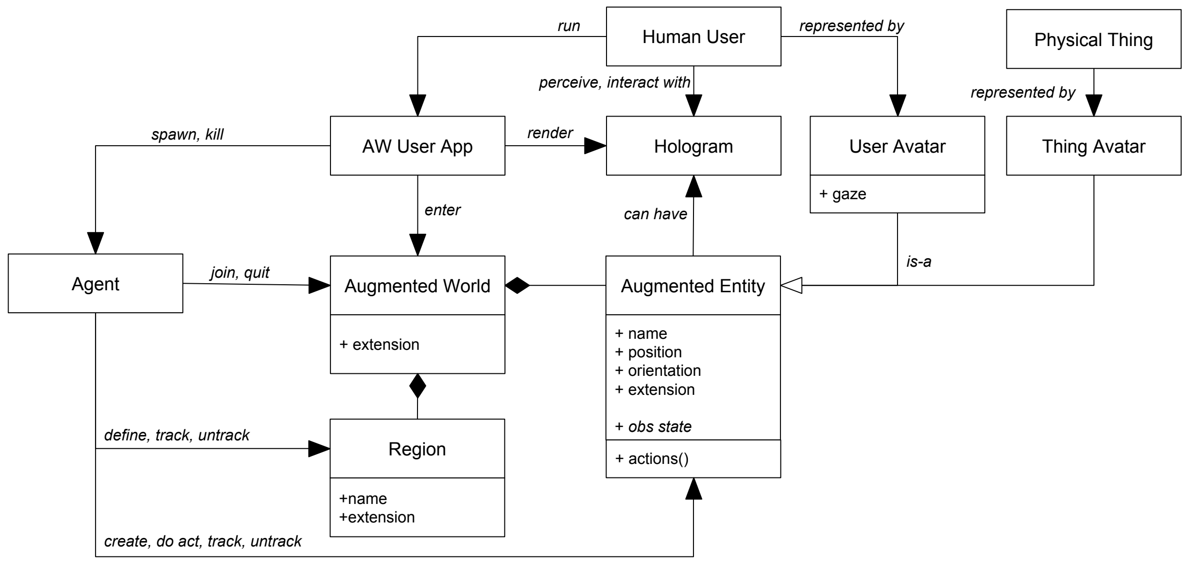


Figure 5.2: The Augmented World conceptual model formalized in UML.

kinds of computational artifacts organized in workspaces exposing observable properties and actions that can be observed and invoked. In fact, augmented entities can be conceptually considered as artifacts in A&A, used as the first-class abstraction to design and develop the application environment where agents are situated. Beyond this, the AW conceptual model introduces further concepts and features, detailed in the following, tailored explicitly for pervasive mixed reality systems, such as – for instance – the spatial coupling with the physical reality and the explicit modelling of human users — which are not part of the A&A meta-model. The main elements representing the conceptual model of an augmented world are formalised in the UML diagram proposed in Figure 5.2 and described in the remainder of this section.

5.3.1 Augmented World

The AugmentedWorld concept represents an instance of an AW logically coupled on a specific portion of the physical world and identified by a logical name (see name property).

The coupling with the physical world is provided by the fact that an AW is featured with a built-in reference system able to logically connect a particular location in the physical world, e.g. a GPS location, with the origin of its reference system. Generally, the reference system of an AW can be modelled alternatively using a Cartesian Coordinate System or a Geographical Positioning System. Possibly, also a combination of them can be exploited, supporting both indoor and outdoor scenarios concurrently.

Besides the reference systems, the AugmentedWorld concept is also featured by the extension attribute representing the physical world area covered by the AW. The area could be simply defined as a geometrical shape centred in the origin of the reference system, e.g. a circular area defined with a particular radius, or more precisely using a set of GPS location as extremities of a more complex shape over a map.

The physical world area covered by the AW can be logically split into regions (see the region concept in the model) to identify particular sub-areas. For instance, if we consider an AW for a building, each room or each level can be defined as a region of the AW. Vice versa, considering an AW covering a city, the area of a square or the path of a particular road can be represented as a region.

One or more regions can compose each AW, not necessarily covering the full extension of the AW. Also, two or more regions can be partially overlapped. The region concept is useful to exploit the spatial observability feature provided by the AWs: humans, agents and augmented entities can be tracked/observed alternatively considering their exact location or by the fact that they are or not within the area of a region. A region is featured by a name and an extension. This latter attribute equals to the one related to the AugmentedWorld concept.

Finally, to each AugmentedWorld instance is associated with a set of Augmented Entities, representing the concrete (virtual) computational objects layered on top of the physical world and described in the following.

5.3.2 Augmented Entity

Each Augmented Entity (AE) defined within an instance of an AW has a specific position, orientation and extension in the physical world (see position, orientation and extension attributes in the model), which are defined with respect to the reference system defined by the Augmented World. In this case, the extension is a solid geometry representing the physical 3D space occupied by the representation of the augmented entity.

An important consideration here is that augmented entities are meant to model not just data items – like in the case of POIs (Point-of-Interests) – but any full-fledged computational entity, eventually encapsulating a state and a behaviour. This is the basic fundamental feature which is useful in particular to build applications where the augmentation is not just an information layer about physical things, but more complex services or functionalities.

Each AE, that can be considered in execution in the physical location described by its position attribute, exposes an observable state (`obs_state` attribute in the model) – in terms of a set of observable custom properties, that can change over time. Of course, also built-in properties (the position, the orientation and the extension) can be observed by agents and humans. Observing (tracking) an AE means being notified when a particular attribute/property of the entity is updated. For instance, an agent that wants to react when a particular entity is moved nearby a particular physical location, the agent can track the location attribute of the entity to be notified each time the entity moves to a new location, possibly checking if the new location is nearby the location of interest and so acting accordingly.

Moreover, each AE also exposes an action interface (see actions() operations) that can be exploited from agents to invoke operations over the AE with the purpose to update its internal state. Exploiting actions, Augmented entities can be dynamically created and disposed of by agents, as well as, for instance, moved in different locations covered by the augmented world extension.

5.3.3 Holograms

The Hologram concept gives the *bridge* with augmented reality. Each AE can have a hologram associated: the hologram is the augmented reality representation of the Augmented Entity, to be perceived by the human users equipped with proper devices. This representation – the hologram geometry – can be as simple as a 2D text, or a complex 3D structure.

An important key point is that this representation is anchored (registered, in augmented reality terms) to the physical world considering the location of the associated augmented entity in relation to the augmented world reference system.

The hologram state depends on the corresponding Augmented Entity state and is kept updated every time the Augmented Entity state is updated. For instance, the orientation of a hologram is updated according to the orientation of the related augmented entity. Also, updating the value of an observable property on the augmented entity, in case this property is used to model a particularity of the associated hologram (e.g., the colour) the information about the property new value also affects on the related hologram.

The hologram view – i.e., the actual rendering in the AR view – depends on the hologram state and considers specific capabilities offered by the device used by users. In particular, the conceptual model allows having more than one geometry associated with each hologram to support several degrees of augmentation (that is a requirement of the augmented world vision). In fact, in case of different human users enter into the augmented world exploiting heterogeneous mixed reality devices – some more powerful than others – each hologram can propose to each human a different representation of itself according to each device capabilities. To clarify, imagine an augmented entity representing a clock perceivable in augmented reality perspective that having an associated hologram exposing two geometries: a 2D geometry like a text on a label for those devices that can only exploit low quality cameras, e.g. a tablet, and a full 3D geometry for those devices offering a more powerful augmented reality experience, e.g. AR visors. Both geometries would be updated according to the augmented entity state, the current time in this example, but the information would be presented to users in different ways according to their device's features.

Besides its representation, a hologram is what enables the interaction with human users, concerning gesture and gaze. In particular, holograms are featured with ad hoc

Primitive Actions	**Description**
joinAW(name, location): awID	to join an existing augmented world, getting an id of the session
quitAW(awID)	to quit from working in an augmented word
createAE(awID, name, template, args, config): aeID	to create a new augmented entity in a specified augmented world, specifying its name, template, parameters (that depend on the specific template), and initial configuration (including position, orientation, ...)
disposeAE(aeID)	to dispose an existing augmented entity
trackAE(aeID)	to start tracking an existing augmented entity
stopTrackingAE(aeID)	to stop tracking an existing augmented entity
moveAE(aeID, pos, orientation)	to change the position and orientation of an augmented entity, if allowed
defineRegion(awID, name, region)	to define a named region, specifying a name and the extension
trackRegion(awID, name)	to start tracking a region
stopTrackingRegion(awID, name)	to start tracking a region

Table 5.1: The agents APIs for predefined functionalities.

properties to recognize, for example, when a user is looking toward it. These properties are managed as the other observable properties of the related augmented entity. Beside the gaze, other properties enabling the interactions with human users are related the ability to grab augmented entities holograms and to touch them in augmented reality. Finally, considering its lifetime, in the conceptual model we can assume that a hologram exists within the AW – or rather, is perceivable by humans – from the moment when the related augmented entity is created and until agents do not destroy it.

5.3.4 Agents

In order to work inside an Augmented World, an agent has to join it, setting up a working session – more precisely, to receive an authorization token to ensure the possibility to act within the AW and on the existing augmented entities. Once joined an Augmented World, an agent can act upon an Augmented Entity, through the actions that the Augmented Entity makes it available, and track it, to perceive its observable state and events.

Besides tracking specific augmented entities, agents can track regions inside Augmented World, as portions of the physical space on which the Augmented World is mapped. As soon as an augmented entity enters or leaves a region tracked by an agent, the agent perceives a corresponding event, carrying on information about the Augmented Entity source of the event.

The set of actions that an agent can do inside an AW can be grouped into two main categories. The first one is fixed set of primitives providing predefined functionalities (see Table 5.1): to join and quit an AW, to create and manage augmented entities, and to define and track regions. Besides this fixed set, by joining an AW, the set of available actions is given by the collection of all action interfaces provided by the augmented entities actually active in that AW. So given an augmented entity aeID providing some action (operation) op, an agent has an action: doAct(aeID, op, args) which triggers the execution of the operation on the augmented entity. As in the case of environment programming abstractions, actions can be long-term processes whose execution occurs inside the augmented entity, eventually completing with a success or a failure. These action events are perceived by the agent triggering the action asynchronously.

On the perception side, in an augmented world, an agent can track either specific augmented entities (trackAE primitive) or regions (trackRegion primitive). By start tracking an Augmented Entity, an agent continuously receives percepts about its observable state and events generated. By tracking a region, an agent perceives events related to augmented entities entering or exiting, dynamically discovering their identifiers. In cognitive agents based on the BDI model, beliefs about the state of the Augmented World and Augmented Entity are automatically created/updated/removed depending on what an agent is tracking.

5.3.5 Human Users

A human user starts a session inside an AugmentedWorld using a dedicated software application (see the AWUserApp concept in the model). An AWUserApp spawn the agents that concretely can act inside the AW. From an user-interface (UI) point of view, the AWUserApp is responsible for creating an Augmented Reality view, considering the current position and orientation of the human user – exploiting sensors offered by the user device like the camera, the accelerometer the gyroscope.

Besides, this is the part in charge of detecting user inputs (e.g. gazing, gestures, voice commands). Such commands could be targeted towards specific holograms (e.g., grabbing a hologram, gazing a hologram). A hologram is then a source of observable events that can be perceived by agents, tracking the corresponding Augmented Entity.

Multiple human users can be immersed in the same Augmented World, possibly using different AWUserApp.

5.3.6 Physical World Coupling

Some augmented entities can be used to represent physical objects (that are part of the physical environment) in the Augmented World, as a kind of "mirror", so that the observable state of the Augmented Entity represents a model of the (physical) state of the physical thing. In Figure 5.2 this is represented by the ThingAvatar concept, which is a specialisation of Augmented Entity.

The coupling between the two levels – so that, e.g. the observable state of the Augmented Entity is updated as soon as the physical one changes – is realised by proper augmented entity drivers (similar to device drivers), typically exploiting proper sensors and embedded technology. Then, by tracking an augmented entity coupled with some physical thing, an agent can perceive the physical state of the thing—as abstracted by the Augmented Entity. Among the physical things that can be coupled to an Augmented Entity, an important case is given by the *physical body* of a human user, so that agents can track the position of human users. This is represented by the UserAvatar concept, which is a specialisation Augmented Entity like in the case of ThingAvatar.

The coupling between Augmented Entities and physical things can also work in the opposite direction, that is actions on the Augmented Entity can be useful to affect physical things, by exploiting proper actuators. So, in this case, an agent affect a physical thing of the environment by acting on the corresponding Augmented Entity. This feature is mandatory to realise part of the bidirectional augmentation required by the augmented world vision — the conceptual model fully supports that if we also consider the human representation in the augmented world, the possibility to exploit holograms and the presence of proactive agents able to act on the augmented entities.

As a very dummy example, consider the case where a real light within a room has to be turned on automatically when a human user enters in the room. Using the abstractions provided by the AW conceptual model, the room can be modelled as a region of the AW and the human can be represented through the UserAvatar concept. An agent can track the region to be notified when the user (its avatar) enters the region and, assuming that also the light has been associated with an augmented entity trough the ThingAvatar concept, the agent can exploit actions provided by the entity to turn on/off the light according the presence/absence of humans in the room. Moreover, another agent could observe the status of the light to keep track of the amount of time the light is turned on and calculate, for instance, the energy consumption.

Although it is an elementary example (see Chapter 6 for more relevant application domains and examples) the proposed one demonstrate how exploiting AW abstraction it is possible to merge digital world and physical one allowing a bidirectional interaction beyond possibilities provided, e.g., by the standard IoT. Moreover, assuming the need to allow a human user to turn off the light when he/her is within the room, a button can be modelled as another augmented entity with an associated hologram. In this case, the user perceiving the virtual button exploiting an AR device could gaze or touch the button and an agent observing the button augmented entity can act accordingly to turn off the light.

5.4 Interactions within Augmented Worlds

To better clarify the usage of abstraction introduced by the conceptual model, this section proposes some example of significant interactions that can be realized using the proposed model. These examples are not meant to be exhaustive covering all possibilities, but they represent a taste of what can be obtained in terms of interactions within an environment modelled as an augmented world. Also, these examples are also useful to reason about the differences between the augmented worlds approach and a pure IoT scenario.

Interactions in pure AR style

Figure 5.3 shows some relevant examples of interactions in pure AR style. The case depicted in Figure 5.3a is about an agent acting on an augmented entity MsgBoard through an action available in the augmented entity interface, setMsg (step tagged as 1), changing its observable state. As a consequence, the hologram model associated with the augmented entity gets updated (step 2). This results in changing the hologram views perceived by the human users immersed in the same AW (step 3). Conversely, the case in Figure 5.3b is about agents tracking a VirtButton Augmented Entity (step 1). A human user interacts with (e.g., touches) the hologram view (step 2), generating an event which is propagated to the augmented entity (step 3), updating its observable state. This change is then perceived by all agents tracking the augmented entity (step 4). Finally, the case in Figure 5.3c is a variant of the one in Figure 5.3b, where the human user gazes the hologram of an augmented entity, and this is observed by the agents tracking the augmented entity itself.

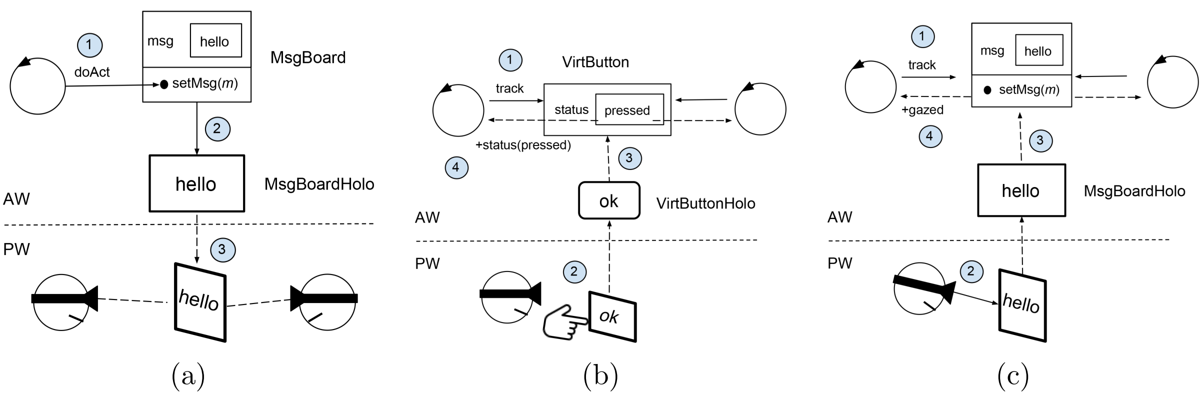
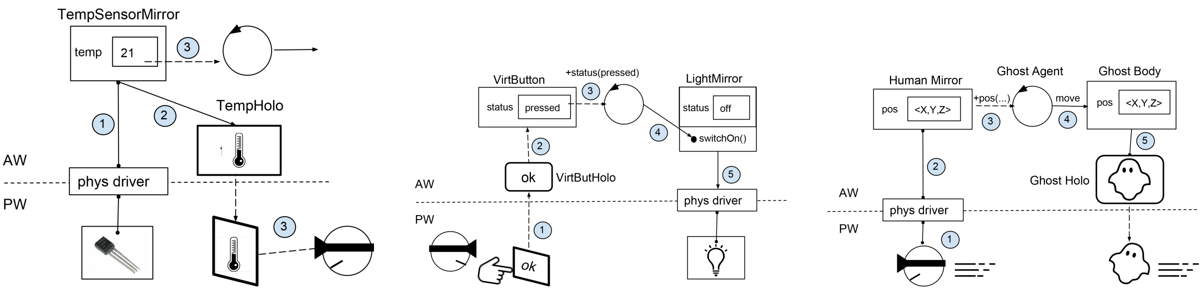


Figure 5.3: Interactions within an AW in pure AR style.

These examples illustrate how the conceptual model proposed naturally support the AR interactions, offering behind appropriate abstractions to manage holograms as computational elements exploiting augmented entities. Using available augmented reality framework, this kind of interaction has to be built by the programmer for each software application, defining a protocol and a strategy to enable software agents (or autonomous entities) to update holograms in a shared and multi-user environment.

Interactions involving Physical Things

Examples proposed in Figure 5.3b include the interaction with physical things. The case in Figure 5.4a is about a TempSensMirror augmented entity coupled to a physical thing (a temperature sensor), featuring also a hologram. The mirror thing has an observable property temp which is kept updated by the driver coherently with the value of the physical sensor. Then, each time the observable property is updated (step 1), the hologram model is updated and human users perceive an updated hologram view (step 3). Besides, also agents tracking the augmented entity can react to the change. The case in Figure 5.4b is about a human user acting on a physical thing (a lamp) indirectly through an augmented entity (virtual button). In this case, an agent tracking the Augmented Entity reacts to the change of state (status becomes pressed) and does an action (switchOn) on an augmented entity functioning as the mirror of the physical thing. Finally, the case in Figure 5.4c is about a hologram following a moving human user. A GhostAgent is tracking the changes of the user avatar related to the human user. As soon as it perceives that the user moved, the agent moves the GhostBody augmented entity, and the hologram model is updated consequently, as well as the hologram view perceived by the user.



(a) (b) (c)

Figure 5.4: Interactions within AW coupled with Physical Things.

The AWs conceptual model proposes a way to interact with physical things that identical to the way used for interacting with digital ones. Humans and software agents can interact with augmented entities regardless of their physical or digital nature. Is the AW that is responsible for keeping connected/coupled physical things to their digital counterpart exploiting ad hoc drivers, eventually but not necessarily running on the physical thing itself – in case it has computational capabilities. In this way, the interaction in AWs abstracts from basics communicational aspects, providing a homogeneous and an interoperable way to manage all elements involved in the running instance of the Mixed Reality environment.

Interactions involving regions

The example in Figure 5.5 involves region tracking. A Majordomo agent tracks a reception region (step 1) and is notified then when a human enters the region (step 2). The current position of the human user is kept updated on the user avatar entity (step 3): when it changes, entering a region currently defined inside the AW (reception), then an observable event is generated (step 4). The agent perceiving the event then acts upon the MsgBoard creating a welcome hologram (step 6).

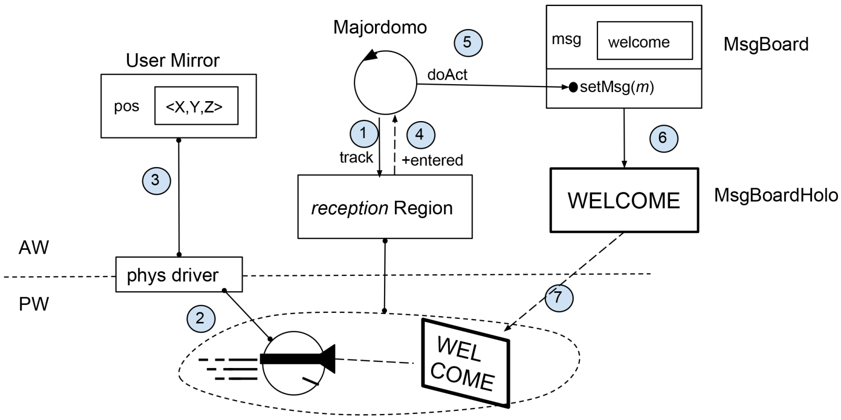


Figure 5.5: Interactions based on region tracking.

Although may seems a marginal concept, regions represent an essential element of the conceptual model. Most of the interactions in AWs are performed exploiting regions. The importance of having modelled the concept of *region* is given by the fact that, for instance, despite an agent can track (and observe) directly a known augmented entity, many are the domains where emerges the exigence to observe and react to a spatial logical region instead of tracking specific elements. This is a feature that is very close to how interactions are generally performed in the real world. Many actions are performed according to perceptions obtained from observed areas where it is not possible to identify a priori the involved entities.

Chapter 6

Application Domains and Relevant Examples

In the previous chapter, the vision about Augmented Worlds has been introduced, providing a concrete definition, describing its rationale and the challenging requirements, and finally discussing the main entities and concepts composing the AW conceptual meta-model. The main aim of this meta-model is to provide the appropriate level of abstraction introducing concepts like augmented entities that can be used for the design of pervasive mixed reality systems integrating autonomous and proactive software components with human users of the system. These latter considered not as pure external users but as active/interactive elements that are part of the system itself. Assuming the proposed conceptual meta-model, we can argue that an AW defines an ecosystem of heterogeneous elements belonging to both the physical reality and the computational/digital one. That is: an ecosystem supporting bidirectional augmentation for the real-time cooperation between those two realities and the collaboration and enhancement of its inhabitants and the cyber-physical environment. The purpose of this chapter is to consider relevant areas and application domains to be exploited to discuss the applicability of the proposed model and obtained advantages concerning design for the road opened by augmented worlds vision towards the building of future smart environments.

6.1 Next Generation of Smart Environments

Undoubtedly, in the short future, the physical environment where people live and work will become smarter than now. Augmented Worlds could represent an answer to the question of how to conceive those new smart environments. Of course, the "smart" prefix could have a lot of means, and undoubtedly Augmented Worlds cannot be the right way to follow to design all categories of smart environments. For example, if we consider a smart environment where the main exigence is to collect data and information about ongoing activities in the physical world – e.g., the status of the road traffic within a city – the augmented world vision probably is not adequate. There are many contexts, instead, where the vision of AW could be applied.

A main lack in what we currently call smart environments is to avoid considering humans parts of the environments, treating them just as users of the systems. For achieving user-centric aware Internet of Things [40] – which however is already a reality, with a vast number of Internet connected objects and devices that has exceeded the number of humans on Earth – that brings together people and their devices into a sustainable ecosystem, first, it is necessary to deal with the integration of disparate technologies, ensuring trusted communications, managing the huge amount of data and services, and bringing users to an active and concrete involvement. Moreover, also those environments where the human presence is more taken into account, humans have not the right elements to interact with the digital environments and among themselves mediated by technologies. Finally, in most of the current systems trying to build smart environments, there is almost the absence of proactive components able to observe and act not only within the digital context but also considering the real one where humans are involved too.

The Augmented Worlds conceptual model could be considered the reference to design novel smart environments where digital and physical coexists within a homogeneous Mixed Reality context.

6.2 Examples of Applications Domains for AWs

This section aims to give a taste of some application domains where the proposed Augmented Worlds conceptual model could be applied to design a new generation of smart environments and to improve people lives and their work styles through digital features and opportunities offered by AWs according to needs of each application domain. Of course, this section is not exhaustive: many other application domains could be considered for augmented worlds. Nevertheless, the ones reported here should be considered as the main areas where nowadays AWs could become very relevant and so where firstly they could be applied.

6.2.1 Industry

Undoubtedly, the industry is the context where nowadays many investments converge with the purposed to innovate industrial processes and to make them smarter than how currently are.

In Section 1.5 we have introduced the actual phenomenon called *Industry 4.0* where augmentation technologies are currently entered in the productive industrial loop based on the idea of Digital Twin — conceived outside of the industrial context and only later considered as a pilaster of the new industrial revolution. However, in the digital twin vision, data and information collected from the real-time industrial process are mainly meant to be processed/accessed for control and statistical purposes only and from actors not directly involved in the industrial process. Also, the digital twin does not directly consider the introduction of Mixed Reality as a medium of interaction with the digital representation of the industrial process. This because the digital twin is built and keep updated in real-time but is not be intended to be used by actors involved in the production.

In this context, the augmented worlds proposal could have a deeper impact on the digital twin vision revolutionizing industrial processes. Those processes are nowadays intrinsically modelled as pervasive software systems: many sensors are involved, and data and information coming from plenty of sources spread within the farm are collected. In Industry 4.0, the connected ecosystem of the Internet of Things (IoT), helps to provide manufacturers and consumers with increased automation, improved communication and monitoring, along with self-diagnosis and new levels of analysis.

By the way, augmented worlds would be able to introduce many others innovative opportunities, for instance:

- use the Mixed Reality approach to let workers receive real-time information from the digital twin, exploiting them to improve their work and productivity;

- introducing proactive agents for real-time observing the industrial process to notify workers (and potentially also other actors) about problems and critical issues exploiting, e.g., mixed reality to put alters in the digital environment;

- let workers to be involved in a cooperative digital industry, to exploit augmented features as a team and not only as a single unit, to both use the mixed reality medium and personal software agents collaboratively to inform each other about the ongoing work;

- helping for reducing the gap between asset inspection and asset maintenance with mixed reality opportunities offered by AWs. The traditional requirement in the industry for a physical inspection can be reduced or even eliminated because products conditions can be fully understood without the need to act physically, offering to the expert the possibility to take corrective maintenance actions earlier in time.

Admittedly, these features could be obtained similarly, also developing ad-hoc software systems as a kind of "plugins" to existing ones. Despite this, augmented worlds bring a new way of conceiving smart industrial environments and not only a way to extend existing industrial processes. So the main benefit is related to the fact that, using AWs, factories will become increasingly automated and self-monitoring as the machines within are given the ability to analyse and communicate with each other and their human co-workers, delivering much smoother processes, and possibly, freeing up workers for other tasks and activities.

6.2.2 Healthcare

The healthcare area represents a very challenging domain of interest for Augmented Worlds, from their study to their application.

On the one hand, it is an area where the introduction of novel technologies such as Mixed Reality and/or Pervasive Computing could have a deep impact in performing healthcare activities by physicians, not only limited to patients assistance. On the other hand, however, healthcare is probably the most critical domain where to introduce innovative technologies, because it covers a lot of different areas with plenty of different features and requirements, sometimes in contrast among them. In other words, healthcare is a domain where tradition often clashes with innovation or where the introduction of the latter proceeds very slowly if compared to other domains/areas.

Despite this, we can suppose in the short future the innovative technologies will assume a significant role in healthcare, for supporting and enriching healthcare tasks and processes.

For what concerns AWs, applications in healthcare can be many and ranging from physicians and nurses training in Mixed Reality environments towards a vision where next generation of hospitals could be designed as "augmented hospitals".

Medical Training and Education

In medical training and simulation, augmented reality provides rich contextual learning for medical students to aid in achieving core competencies, such as decision making, effective teamwork and creative adaptation of global resources towards addressing local priorities. Moreover, the patients' safety is safeguarded if mistakes are made during skills training with augmented reality.

Medical training can be interactive or passive, solitary or cooperative, referred to real patient data/information or not.

Augmented Worlds for medical training could be used to build a hybrid environment where a team of physicians or medical students could concurrently experiment medical techniques assisted by technology offering to them features such as, e.g., alerts in relation to the real-time context or the possibility to have real-time data and information from real patients to be used in the simulated environments, and so on. A mixed Reality environment is undoubtedly a better learning environment if compared to a completely virtual one. In the first, in fact, the physician stays in the physical world but exploits actively or passively the technology to improve his/her competencies and to better accomplish his/her tasks.

Tracking and Assisting activities

In healthcare, documenting carried out activities and followed processes during patient care and related treatment represents one of the most critical activities. In particular, for keep track of history for the patient, for an analysis of the performed work in relation to obtained results, for measurements purposes and costs analysis, and so on.

Usually, health documentation is something that is produced a posteriori, at the end of the related medical process and, sometimes, with a significant delay. For instance, medical surgery or emergency treatments require to the team of doctors involved, in most cases, to produce documentation of all the operations performed, e.g., of the drugs administered and anything else about the progress of the patient's state during the treatment. Because of the rapidity of performed operations in these contexts and supposing that the concentration of doctors is completely addressed to the current health procedures, is unthinkable that the team compiles contextually with the operations also a sort of diary as documentation in the absence of a dedicated employee to do this. Obviously, the documentation it could be only compiled later on by retracing the memories of the medical team and then in an often inaccurate and summary manner.

The opportunities introduced by augmented world vision could help to improve several medical tasks with the introduction of personal agents able to observe the ongoing medical activities, taking notes of performed procedures and tracking also the patient's medical real-time status. Personal agents living in the augmented world, shaping the rooms (hospital areas) where medical operations are carried out, could exploit environment sensors and medical machinery to learn information firstly to observe, but also to alert the team about critical issues for the patient and to suggest them several solutions in some cases. In this case, the mixed reality dimension offered by AWs is not required to be used, although it could help medical actions in a future vision of hospitals (here it will be important to discuss about mixed reality effects on physicians, and medical environments, researching the best way for avoiding to distract them from their activities and so on).

This application domain for tracking and assisting in healthcare steady the applicability of the vision of augmented worlds highlighting the fact that, potentially, augmented worlds could also be useful without the mixed reality dimension, but only with the opportunity to introduce cognitive agents into the hybrid environment for observing the physical environments and involved humans' activities with the ability to act upon both physical and digital layers.

Towards Augmented Hospitals

Considering reported considerations in previous paragraphs of this section, a challenge for the application of augmented worlds can be identified in the design of future smart augmented hospitals to offer new opportunities in patients care and to improve medical tasks and activities. With the introduction of augmented world vision in the hospital context, plenty of new form of applications could be identified and developed. Many companies are currently working in the direction to make smarter hospitals, but again (as said for the industrial context in the previous section) the augmented world vision could help to rethink to the entire process of patient care and not only to extend what is currently available in hospitals. So, bringing to better results in terms of the importance of the produced software, usability and applicability. Also, they can have a deep impact on the design of future hospitals, starting from buildings design and medical processes definitions and implementations. In this case, the challenge is hard to be dealt with, but the effects of AWs in this sector could be very disruptive. For this reason, the healthcare context represents a significant area where investigate and design augmented worlds for the next generations.

6.2.3 Cultural Heritage

Cultural Computing (CC) is an emerging field that applies computer technology and scientific methods to culture, arts, and the social sciences to represent, enhance, extend, and transform creative products and processes. Since the mid-2000s, the use of enabling technologies in Cultural Heritage has been extended to immersive technologies – like Mixed Reality oriented technologies – which provide sensory experiences through various combinations of real and digital content. Cultural Heritage benefits significantly from the use of these technologies. Users can experience cultural artefacts in a completely new way. Some studies demonstrate the viability of Mixed Reality adoption for different application areas in Cultural Heritage. Nevertheless, analysing most of these studies emerges that in almost all the introduction of MR is related to proposing a single user experience, sometimes without an effective interaction of the user with the computer generated holograms.

Considering the vision about augmented worlds, Cultural Heritage could represent a very interesting application domain for it. Museums, archaeological sites and many other artistic/historical contexts could be coupled with an ad-hoc designed augmented world to propose to users and visitors new amazing ways to interact with, e.g., art elements (paintings, sculptures, and so on) and historical artefacts or environments.

On the whole, cultural heritage sites and artefacts gain significant added value from enrichment through digital media. Moreover, the presence in a cultural heritage augmented world of proactive software agents could enrich the cultural site not only in terms of new ways of visualizations but also concerning functionalities for monitoring, observe and maintenance culture. For instance, consider the opportunity to digitally connect two separated museum, to show in the first digital twins of artefacts physically available in the other museum. Also, the possibility for agents to propose dynamic information for users moving in an archaeological site according to their observation of the physical environment, guiding visitors or let them know more information according to their interests.

Later in this dissertation, a case study about an augmented world designed for an *Augmented Museum* will be discussed. Other case studies in this direction are currently under development within outdoor contexts.

Part III

The **MiRAgE** Infrastructure

Chapter 7

Engineering Augmented Worlds

The conceptual model for Augmented Worlds proposed in Chapter 5 defines relations among elements and concepts that can be used as abstractions to design pervasive mixed reality (agent-based) smart environments. Although all main features of an AW are taken in account by the model – which proposes all abstractions/concepts that can be used as a base for their design – there is still an infrastructural gap to be filled to make effective the design and the implementation of such complex software systems as AWs are. In a sense, moving from "theory" to "practice" requires to consider many other issues – mainly related to involved technologies – that should be addressed from an infrastructural point of view to proposing to developers a concrete software framework to be used to engineering the application logic of each instance of augmented worlds. The goal of this chapter is to introduce the second main contribution of this dissertation that is related to the engineering of a concrete framework/platform – called MiRAgE – for developing Augmented Worlds.

7.1 Introduction to MiRAgE

To concretely support the design and the development of Augmented Worlds, in this dissertation, we introduce the MiRAgE framework[a] – where the name is an acronym for *Mixed Reality Agent-based Environments*. According to the defined Augmented World conceptual framework, MiRAgE allows designing, develop and deploy augmented worlds instances. In particular, the framework provides an API and a runtime to develop and run augmented worlds, also providing a high-level abstraction to integrate agents and MAS in pervasive mixed reality complex systems.

MiRAgE is designed to be used by a software developer that wants to build an augmented world abstracting from all infrastructural issues emerging when dealing with

[a]This framework was presented in the demo session of the 16$^{\text{th}}$ International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS) in 2018 [20, 19].

pervasive and distributed systems and avoiding of taking in account as much as possible required Augmented Reality platforms and libraries. In other words, the main aim of MiRAgE is to allow the AW developer to build such a complex system to avoid starting from scratch but mostly focusing on the specific application logic. In particular, the developer of an AW instance should focus only to design the involved augmented entities – designing their internal state in terms of observable properties, provided actions and related hologram geometries – and coding the behaviour of agents acting within the augmented world.

7.1.1 Requirements

Discussing functional requirements that MiRAgE should satisfy, firstly we can surely assert that the framework has to implement the AW conceptual model, effectively providing all theoretical abstractions introduced by the model. In particular, MiRAgE has to provide an effective way to quickly design augmented entities and an interoperable mechanism to allow agents and humans to interact with the augmented world.

All coarse-grained features discussed in Section 5.1.2 for Augmented Worlds needs to be taken into account by MiRAgE, as well as the framework has to provide also satisfactory solutions for challenges addressed in Section 5.1.3. For the ease of the reader, the following list briefly recaps those features and challenges.

- Support a multi-user context and cooperation among humans and for teams. Also, proactive software agents must have the same capabilities as humans, especially in term of collaboration.

- Offer an interactive context where each user can assume both an active or passive role respect to the augmented world's entities in terms of interactions (acting proactively or being notified or observed considering needs).

- Allow to each augmented world instance to be accessed in (soft) real-time as a shared and distributed environment for humans and agents.

- Define proper support for a *bidirectional augmentation* between the physical layer and the digital one. Each layer has to be informed and updated considering the status' evolution over time of the other layer.

- Propose a "complete" mixed reality experience, allowing holograms to have complex behaviours and complex interactions with humans.

- Efficiently deal with the causal consistency issue, offering also a solution for managing the unpredictability of events due to the close relation of each augmented world instance with the real world (mostly unpredictable in terms of events that can occur).

Besides, the following paragraphs discuss another three relevant requirements MiRAgE must satisfy.

Combined support for Indoor and Outdoor scenarios. Develop a Mixed Reality software system for an indoor scenario rather than an outdoor one implies to take into account different aspects especially for what concerns the registration of augmented elements with the real world. In an indoor scenario, an approach based on GPS (Global Positioning System) localization should be avoided due to the inaccuracy of the GPS inside buildings. Marker-based approaches and techniques of ambient mapping should be preferred. Vice versa, in an outdoor scenario, sometimes GPS localization represents the best approach to understand where the user is and to register holograms with the real world, especially when there is the requirement about covering a wide area. MiRAgE should ensure the augmented world approach would be effective in indoor scenarios as is in outdoor ones. In particular, the reference systems of the AW should contemplate both scenarios concurrently, allowing for developing AW where different mechanisms for holograms registration can be combined. For instance, an AW could cover an area where the GPS localization is used for defining the registration of holograms with a coarse-grain precision combined with a marker-based approach to be used selectively when a user application detects one of them to provide a finer precision.

Support for multiple agent technologies. From the agent side, not a particular technology for developing cognitive agents should be considered in MiRAgE. As requested by the model, the framework should propose interoperable APIs to be exploited by different agent-oriented languages and framework. MiRAgE framework should be interoperable in terms of agents languages used to develop cognitive agents that join and interact with the running augmented world selected instance.

Support for heterogeneous AR-oriented devices. Augmented Worlds are dynamic environments where agents and humans dynamically join/enter and leave the environment. In such a scenario, it is not possible to assume that a user has a particular AR enabling device and must be supported the possibility for different users to enter to the AW with different devices – from the simplest, like smartphones and tablets, to wearable see-through devices, such as smartglasses and AR visors (refer to Section 2.4 for more details). For this reason, the component of MiRAgE devoted to supporting humans to see and interact with holograms should be designed to well adapts to devices exposing different features. In other words, MiRAgE must ensure a way for humans to join the AW and see holograms with different degrees of augmentation related to their own devices. For instance, each hologram representation could be different in relation with computational power offered by the involved device used to render the augmented reality content.

7.1.2 Design Principles

An objective of the MiRAgE infrastructure is to become a reference platform to develop agent-based pervasive mixed reality environments. The Augmented World conceptual model has been defined in order to reach this objective from the abstraction level. Nevertheless, is important for MiRAgE to consider also the available technological panorama and to deal with enabling existing technologies, promoting the reuse of competences in Mixed Reality framework or agent-oriented languages possibly owned by the developers that will use the framework to build augmented worlds. Considering AR enabling technologies, the design of the framework has to allow for fully reuse and exploit existing AR technologies, both software and hardware, keeping a strong separation by the enabling level and the agent/humans level.

A first important design guideline is represented by *generality*. In particular the framework aims at being sufficiently general to support the development of effective AW for different application domains exposing heterogeneous features and requirements.

Secondly, all design principles defined in literature both for distributed software systems and mixed reality systems must be taken into account. In particular, the *reliability* of the systems is important as well as the *adaptability* to changes of configuration both in terms of involved agents/humans and openness of the approach.

Concluding, MiRAgE must guarantee also a *scalable approach*. Entities involved in the augmented world are logically distributed within the real environment but is not important where they are actually running. In other words, even if an augmented world is a distributed system, from an infrastructural point of view, all entities composing populating the digital level could be in execution on a single computer. That being said, MiRAgE must contemplate an environment populated with a huge number of augmented entities that could require a distribution of the computation over several computing nodes. Moreover, augmented worlds are systems where also multiple agents and multiple humans join and interact. This requires to keep active lot channels for communication, to notify them and react to agents/humans real-time requests. Summarising, from a logical point of view each instance of an augmented world should be considered as a monolithic software system running on a single computational node. From a practical point of view, instead, MiRAgE should contemplate a distributed execution, possibly exploiting also cloud-based solutions. Anyway, the runtime scalability of the infrastructure or the distribution of the computation must be invisible both for human users and for involved software agents. It is a task of MiRAgE to manage computational entities to make them accessible regardless of the underlying infrastructure.

7.1.3 Main use cases

According to the augmented world meta-model, MiRAgE has to achieve to several uses cases. Figure 7.1 reports the main ones. In particular, for main involved actors of the

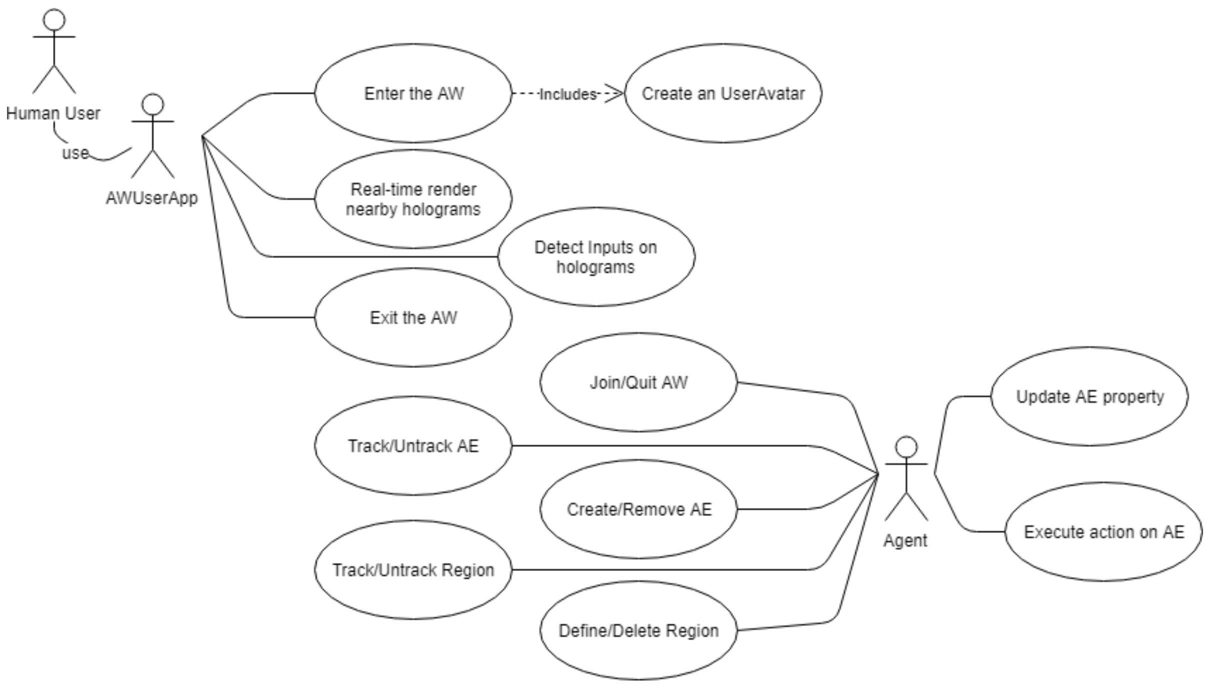


Figure 7.1: MiRAgE relevant use cases for involved actors.

system – human users mediated by the AWUserApp concept and agents – respectively has to enter or join the augmented world instance before starting to acting within. In Figure 7.1 has been only an extract of all MiRAgE use cases, i.e. are not reported ones related to physical things. Also, it is supposed that the instance of an augmented world exists before agents and humans start interacting and collaborating in the digital environment shaped by the augmented world.

7.2 Logical Architecture

Considering reported requirements and objectives, a logical architecture for MiRAgE has been designed. In Figure 7.2 there are represented the three main components of this architecture. In particular:

- the AW-Runtime, which provides the infrastructure to execute AW instances, managing the augmented entities execution and providing a standard interface for agents to interact and observe them;
- the HologramEngine, which supports holograms visualization and keeps updated their geometries in all human users devices, considering physics and managing inputs/actions of users over holograms;

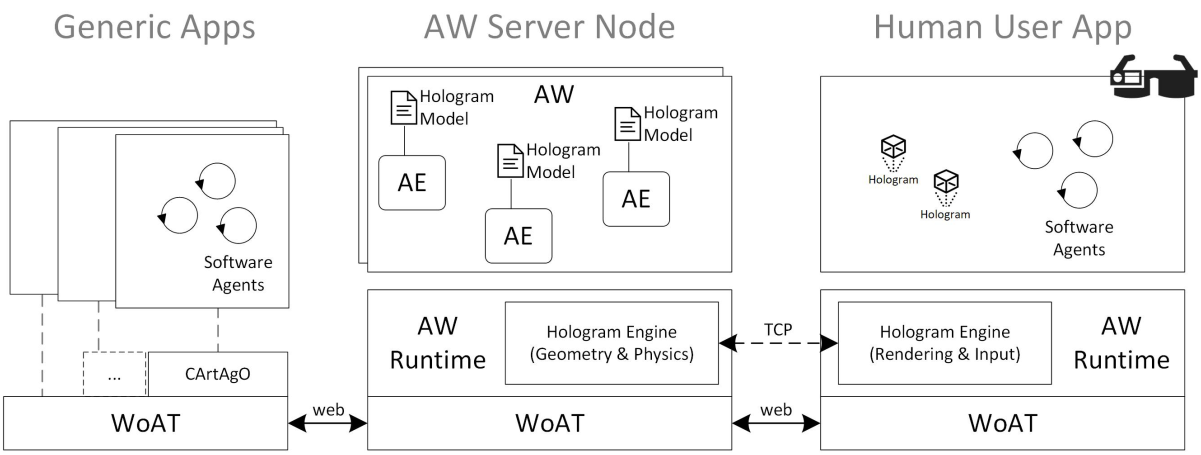


Figure 7.2: MiRAgE Logical Architecture.

- the WoAT layer, which enables full interoperability at the application level.

Following paragraphs will describe each (macro)component of the MiRAgE logical architecture from a structural point of view, also considering the interaction with other elements.

7.2.1 The AW-Runtime component

The AW-Runtime represent the main component of MiRAgE – the "heart" of the infrastructure – acting as a kind Virtual Machine for executing an instance of an augmented world. In particular, the runtime is designed to run on a server node (possibly distributed in cloud architecture), hosting the execution of one or multiple instances of augmented worlds.

Logically, it represents the place where augmented entities are in execution, where the state of the augmented entities is stored and kept updated. More in details, it represents the node to be contacted from agents to interact with augmented entities and has the responsibility to provides all mechanisms to be used to manage augmented entities and interact with them. For instance, a task of the AW-Runtime concerns the generation and consequently propagation of all events related to the running instance of the AW and its involved entities. Each agent refers to the AW-Runtime to retrieve the APIs to be used to acts over augmented entities, regions and so on (see details about the WoAT layer for more details). Moreover, is under the responsibility of the AW-Runtime to notify to interested agents updates of specific augmented entities internal status of observable properties.

Because it knows all AWUserApps connected – according to the model each AWUserApp need to enter to an AW if it wants to see holograms and receive updates for their rendering – is the AW-Runtime the component of MiRAgE responsible also for informing each HologramEngine connected about each hologram creation or dispose – contextually to the creation or dispose of related augmented entities – and about each update to running holograms' model.

Finally, it has the responsibility to keep aligned and updated also physical things involved in the AW running instance as ThingAvatar.

7.2.2 The HologramEngine component

The main task of this component is to keep holograms (their geometries and properties) updated and consistent in each user application that wants to interact with the AW. It wraps and exploits enabling MR and AR technologies. It runs both on the AW-Runtime node and each user device, directly communicating, e.g., through TCP channels.

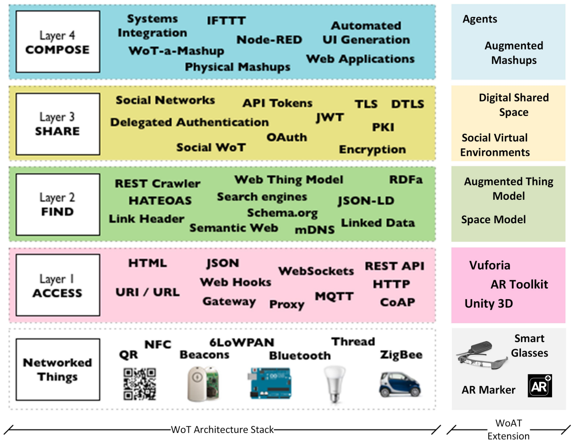
From the AW point of view, the HologramEngine provides the support to design the geometry of each hologram and keeps updated its representation according to entities properties – the part of the AW-Runtime hosting a portion of the HologramEngine is responsible for this.

From the user perspective, it provides all features needed to display the hologram according to the Augmented/Mixed Reality view. It is responsible for managing physics (e.g. collisions between two holograms or perspective issues between holograms and real objects) and to manage user inputs and interactions with holograms (e.g. allowing to exploit gaze as a form of interaction or detecting hands/fingers gestures to manipulate holograms).

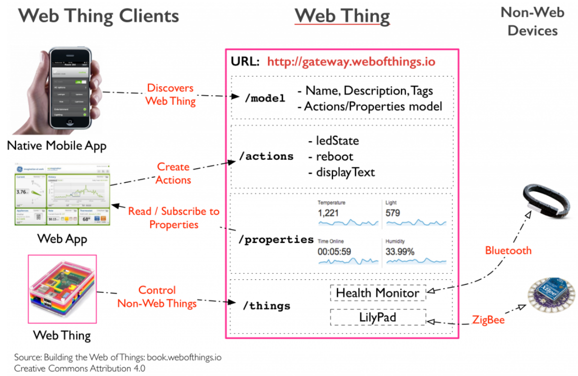
7.2.3 The Web of Augmented Things (WoAT) layer

The AW vision – and so MiRAgE – promotes the design and development of smart environments as multi-user, distributed and open systems. Like in pure IoT systems, also in the AW perspective *interoperability* is a significant issue that has to be appropriately managed.

Making MiRAgE an interoperable infrastructure means allowing the join and the interaction with the infrastructure from external entities as simple as possible effortlessly. For example, agents developed with different programming paradigms should be able to join and act within the AW in the same way. Moreover, external applications that want to interact with a runtime instance of an AW must be able to do so without defining a particular communication protocol or dealing with specific ad-hoc requirements. The idea to deal with this important requirement for MiRAgE is to exploiting the well know Web-of-Things (WoT) model and principles [33] by defining an interoperable application layer, able also to extends by itself the WoT with Mixed Reality oriented capabilities (see Figure 7.3a). In this dissertation, we refer to this layer with the name of *Web of Augmented Things* (WoAT) [18, 17].



(a) WoT stack extended to WoAT. (*figure partially readapted from picture taken in [33]*)



(b) The original Web Thing model

Figure 7.3: Elements supporting the WoAT layer design.

From a design perspective, the WoAT architecture has been conceived by considering the following requirements:

- full interoperability and openness at the application level, to be open to any technology for implementing agents working inside augmented worlds, yet enforcing an agent-oriented level of abstraction at the design level;
- distribution and scalability in the model adopted for defining and structuring the set of augmented entities;
- clean integration with the Internet of Things and the Web of Things "worlds".

From WoT to WoAT

The basic idea is to model every augmented entity, regions, etc. as a *thing* in the WoT perspective, basically featuring:

- a root URL, that corresponds to its network address;
- a RESTful Web API, based on self-descriptive messages, HTTP operations (GET, POST, PUT, DELETE, HEAD) and event-oriented mechanisms (e.g. web sockets).

Besides, the model of augmented entities in the WoAT perspective can be defined as a specialization of the Web Thing model (see Figure 7.3b), so as to include AW's concepts. Accordingly, the root resource model of an AE in the WoAT perspective can be framed as follows:

{ae}	AE root resource URL
{ae}/model/	the model of the AE
{ae}/properties/	the list of properties
{ae}/properties/{id}	a specific observable property
{ae}/actions/	the list of actions
{ae}/actions/{id}	a specific action
{ae}/actions/{id}/activations/{actionId}	a specific execution of an action

Observable properties defining the observable state of the augmented entity are naturally mapped onto properties as defined by the WoT. Analogously, actions (operations), the functions that can be used by agents to act on augmented entities, are naturally mapped onto actions of the WoT model. Then, as in the case of the Web Thing model, also in WoAT we can assume that (1) GET operation can be used to retrieve observable properties of an augmented entity and (2) POST operation can be used to request the execution of an action over an augmented entity. Besides, the augmented entity model includes specific properties, coming from the AW conceptual model, defined in the WoAT as follows:

{ae}/properties/location	location in the physical world
{ae}/properties/orientation	orientation in the physical world
{ae}/properties/extension	extension in the physical world
{ae}/hologram	representation in MR

To support event-oriented interactions, AE in WoAT can be subscribed, both at the resource level and the property/action level, like in the Web Thing model. The subscription allows for agents observing the AE to be notified via a push event whenever the state of an observed resource changes. Besides the augmented entities, also instances of AWs are modelled as a WoT Thing, featuring the following model:

{aw}	AW root resource URL
{aw}/properties/extension	physical extension of the AW
{aw}/properties/entities	list of augmented things
{aw}/properties/regions	list of regions to be tracked
{aw}/properties/regions/{id}	specific region
{aw}/properties/regions/{id}/entities	entities located in that region

The extension property describes the region of the physical world covered by the AW, including the system of reference used for locating the AEs. The entities one describes the dynamic set of augmented entities currently active within the AW instance. Dynamic lookup and discovery of AEs occur by GETs (or subscriptions) on this property, possibly specifying filters. Instead, dynamic creation and removal of AEs occur using POSTs on this property. The regions property is introduced to support a main functionality on the agents' side, which is the continuous observations of a physical region of an augmented world so as to be notified about events related to augmented entities entering/exiting the region. An agent can create a *region* (using a POST on regions) specifying a unique identifier (e.g., room-2) and a physical region (which must be a subregion contained in the AW). Given a new region, a property entities keeps track of the augmented entities that are currently located in the corresponding region. By subscribing to the entities property of a particular region, an agent can perceive when an augmented thing entered the region or left it.

7.3 A first prototype for MiRAgE

According to the designed logical architecture, a prototype[b] for MiRAgE has been developed considering available mainstream technologies.

The AW-Runtime is written in Java, using Vert.x[c] library to implement (part of) the WoAT layer. The HologramEngine is based on Unity equipped with the Vuforia plugin to manage AR aspects and with built-in scripts developed in C# to provides specific features for AW. Currently, a Java-based API is provided for developing the template of augmented entities, allowing to run many instances of the same template. Geometries for holograms are defined exploiting Unity prefabs.

On the agent side, a CArtAgO-based API has been developed to allow the seamless integration of the AW framework with cognitive agent programming languages and platforms supported by CArtAgO [51], such as Jason [9], ASTRA [14] and JaCaMo [8]. Alternatively, the Environment Interface Standard (EIS) [5] approach could be used as well, bridging agent programming technologies and AW ones. Following paragraphs will be devoted to describing how MiRAgE framework has been developed considering the structure and interaction dimensions of main MiRAgE's components.

7.3.1 The AW-Runtime and Augmented Entities

As mentioned, the AW-Runtime has been developed using Java as the main language. The AW-Runtime provides to the AW developer an interface called `Environment` to be used to deploy the AW-Runtime itself and to set up the AW instance. Potentially an AW-Runtime is able to run multiple instances of AWs, but at the current stage of the prototype there is a constraint to run each instance of an AW on a dedicated runtime.

In Figure 7.4 is reported the basic interactions occurring within the AW-Runtime to deploy the runtime and start an instance of an AW. Details about the Hologram Engine Bridge (HEB) and the AWService will be provided later in this section.

[b]The current version of the MiRAgE prototype is open source and is available at the following link: `https://github.com/angelocroatti/mirage`.

[c]Vert.x is a tool-kit for building reactive applications on the JVM (`https://vertx.io/`)

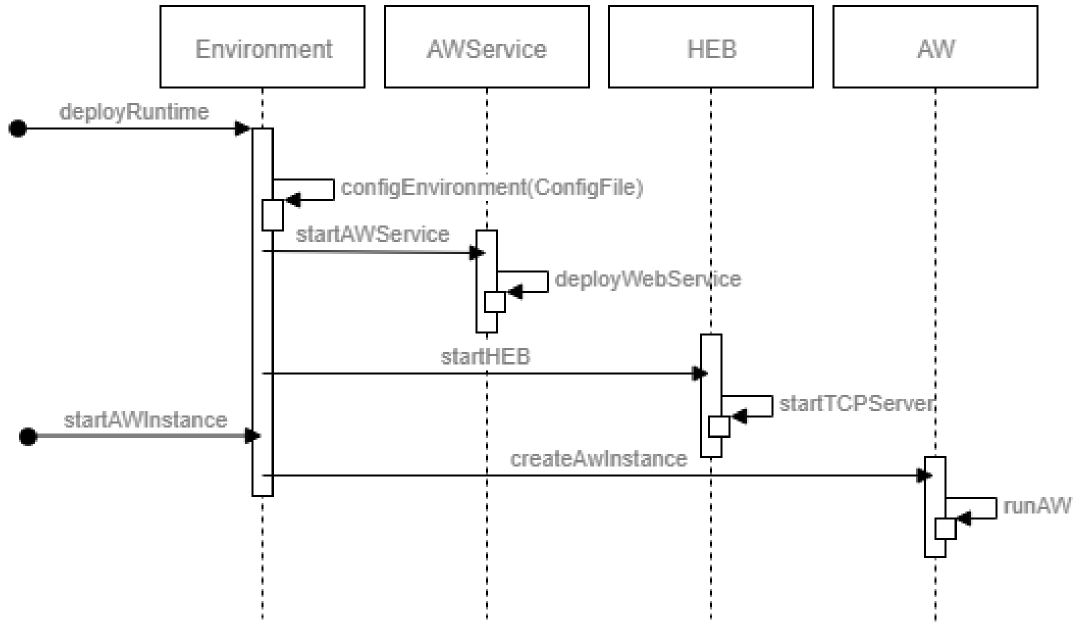


Figure 7.4: Startup interactions in the AW-Runtime for its deployment and the AW instance creation.

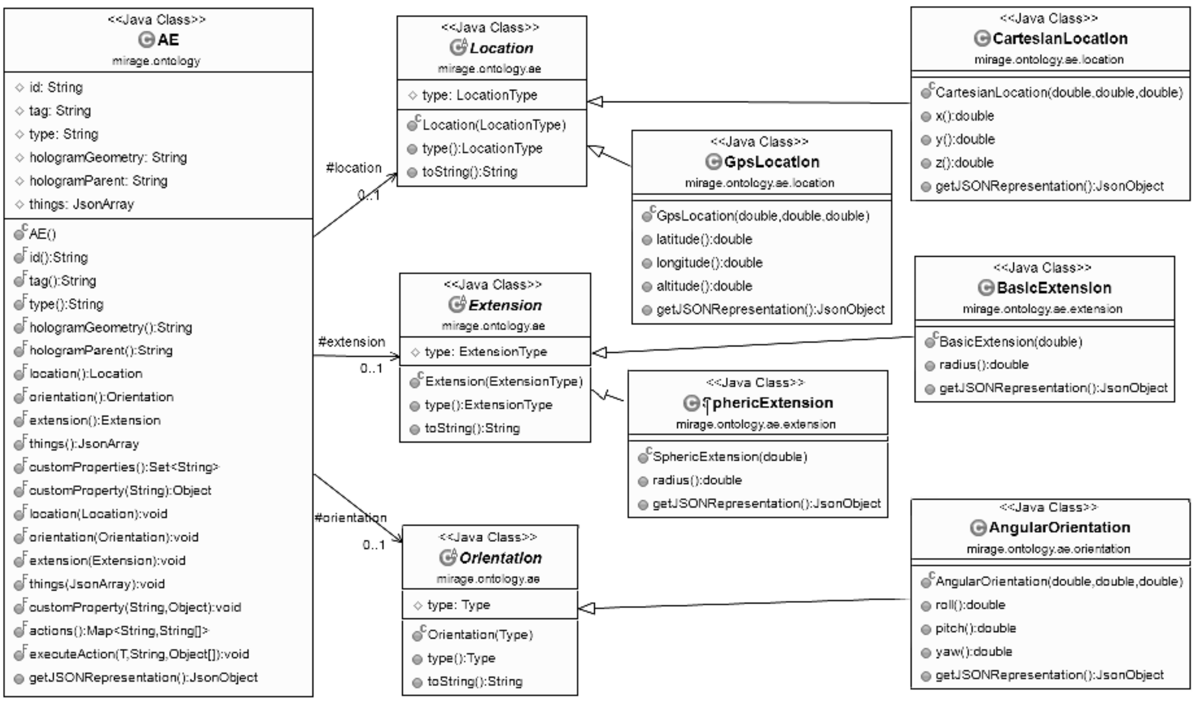


Figure 7.5: The designed ontology for augmented entities in the MiRAgE prototype.

Each AE created within the AW instance is mapped in a dedicated object – in the meaning of the OOP paradigm – created by the AW-Runtime under requests coming from agents exploiting the WoAT layer. The ontology developed for augmented entities is reported in Figure 7.5. Similar ontologies have also been defined for regions and other concepts according to the augmented worlds conceptual meta-model.

7.3.2 The WoAT Service

Considering the definition given in Section 7.2.3 for the WoAT layer, the MiRAgE prototype includes a WebService developed to provide the RESTful APIs of the WoAT layer to be exploited mainly by external agents. Beyond WoAT APIs, the service also provides the possibility to register tracking requests for AE's observable properties and regions through the WebSocket mechanism, as illustrated in the code snippet reported in Listing 7.1, where the setup phase of the service is coded in Java using the Vert.x library. Also, Listing 7.2 reports the WoAT layer APIs exposed by the service in the current state of the MiRAgE prototype development.

```
public class AWService extends AbstractVerticle {

  private Router router;

  @Override
  public void start(final Future<Void> future) {
    router = Router.router(vertx);
    router.route().handler(BodyHandler.create());

    initAPIs();

    vertx.executeBlocking(f1 -> {
            f1.complete(vertx.createHttpServer());
        }, r1 -> {
          final HttpServer server = (HttpServer) r1.result();
          vertx.executeBlocking(f2 -> {
              f2.complete(server.websocketHandler(handleTrackingRequest)
                                    .requestHandler(router::accept));
          }, r2 -> {
              server.listen(Config.WOAT_NODE_SERVICE_PORT, handler -> {
                        manageResult(handler); });
          });
        });
  }

  private void manageResult(final AsyncResult<HttpServer> handler) {
```

```
    final boolean succeeeded = handler.succeeded();

    if(succeeeded) {
      log("Listening on port " + handler.result().actualPort());
    } else {
      log("Failed: " + handler.cause());
      System.exit(0);
    }
  }
}
```

Listing 7.1: A snippet of the Vert.x Verticle implementation of the WoAT layer web service.

```
private void initAPIs() {
  router.get("/service");
  router.get("/info");

  router.get("/aw/:awID/info");
  router.post("/aw/:awID/agents");
  router.delete("/aw/:awID/agents/:sessionID");
  router.post("/aw/:awID/users/sessions");
  router.delete("/aw/:awID/users/sessions/:sessionID");
  router.post("/aw/:awID/users/actions/updateGaze");
  router.post("/aw/:awID/users/actions/updateLocation");

  router.get("/aw/:awID/entities");
  router.post("/aw/:awID/entities");
  router.get("/aw/:awID/entities/:entityID");
  router.delete("/aw/:awID/entities/:entityID");
  router.get("/aw/:awID/entities/:entityID/model");
  router.get("/aw/:awID/entities/:entityID/ model/:modelElement");
  router.get("/aw/:awID/entities/:entityID/properties");
  router.get("/aw/:awID/entities/:entityID/properties/:property");
  router.get("/aw/:awID/entities/:entityID/actions");
  router.post("/aw/:awID/entities/:entityID/actions/:action");

  router.post("/aw/:awID/regions");
  router.get("/aw/:awID/regions");
  router.get("/aw/:awID/regions/:regionID");
}
```

Listing 7.2: The WoAT APIs implemented in the current MiRAgE prototype.

For a more formal definition of the implemented APIs, consider following YAML mapping for the *joinAW* and the *createAE* APIs in Listings 7.3 and 7.4 respectively.


```yaml
'/aw/{awID}/agents':
  post:
    description: Join of a Agent in the AW
    operationId: agentJoin
    parameters:
      - name: awID
        in: path
        description: AW Id
        type: string
        required: true
      - name: credentials
        in: body
        description: agent credentials
        schema: $ref: '#/definitions/AgentCredentials'
        required: true
    responses:
      '201':
        description: 'Successful, join the agent and retrieve session ID'
        schema: $ref: '#/definitions/SessionID'
      '404':
        description: 'Not Found'
      '400':
        description: 'Bad Request'
```


Listing 7.3: The YAML description of the *joinAW* API.


```yaml
'/aw/{awID}/entities':
  post:
    description: Create a new entity in aw
    operationId: createAE
    parameters:
      - name: awID
        in: path
        description: AW Id
        type: string
        required: true
      - name: entityDetails
        in: body
        description: AE details
        schema: $ref: '#/definitions/AEComplete'
```

```
        required: true
    responses:
      '201':
        description: 'Successful, creates the new AE and retrieves its ID'
        schema: $ref: '#/definitions/ID'
      '404':
        description: 'Not Found'
      '400':
        description: 'Bad Request'
      '401':
        description: 'Unauthorized'
```

Listing 7.4: The YAML description of the *createAE* API.

Analogously, exploiting YAML, we defined all other APIs required by the augmented worlds meta-model (Table 5.1 of Section 5.3 reports the complete list).

7.3.3 The Hologram Engine Bridge

The connection between the AW-Runtime and each HologramEngine is managed by a component called Hologram Engine Bridge. Part of the Hologram Engine Bridge is in execution on the AW-Runtime side (coded in Java) while the other part is in execution on each HologramEngine node (coded in C# within the Unity-based App).

The model in Figure 7.6 reports the structure of main classes involved in the Hologram Engine Bridge, both on the server node and each AWUserApp. The Hologram Engine Bridge communication is based on the TCP protocol and the responsibility to connect to the AW-Runtime is in charge of each HologramEngine that need to know the IP address of the server where the AW-Runtime is in execution.

The main task of the Hologram Engine Bridge is about allowing the AW-Runtime to broadcast to each HologramEngine the events related to holograms. In particular, the creation of a new hologram and changes in the properties of existing holograms. For instance, when an agent moves an augmented entity from an original location to another position within the reference system, if the involved augmented entity has an attached hologram, the information about location change is encapsulated in a proper JSON object and spread to all connected HologramEngines. Also, the Hologram Engine Bridge let an HologramEngine to exploits the other direction of its open connection to the AW-Runtime to inform the augmented world about inputs of the user using the engine within is AWUserApp on a particular hologram. Furthermore, even though the Hologram Engine Bridge flows to the AW-Runtime information about infrastructural updates for each UserAvatar in terms of position and orientation considering mobile device sensors updates.

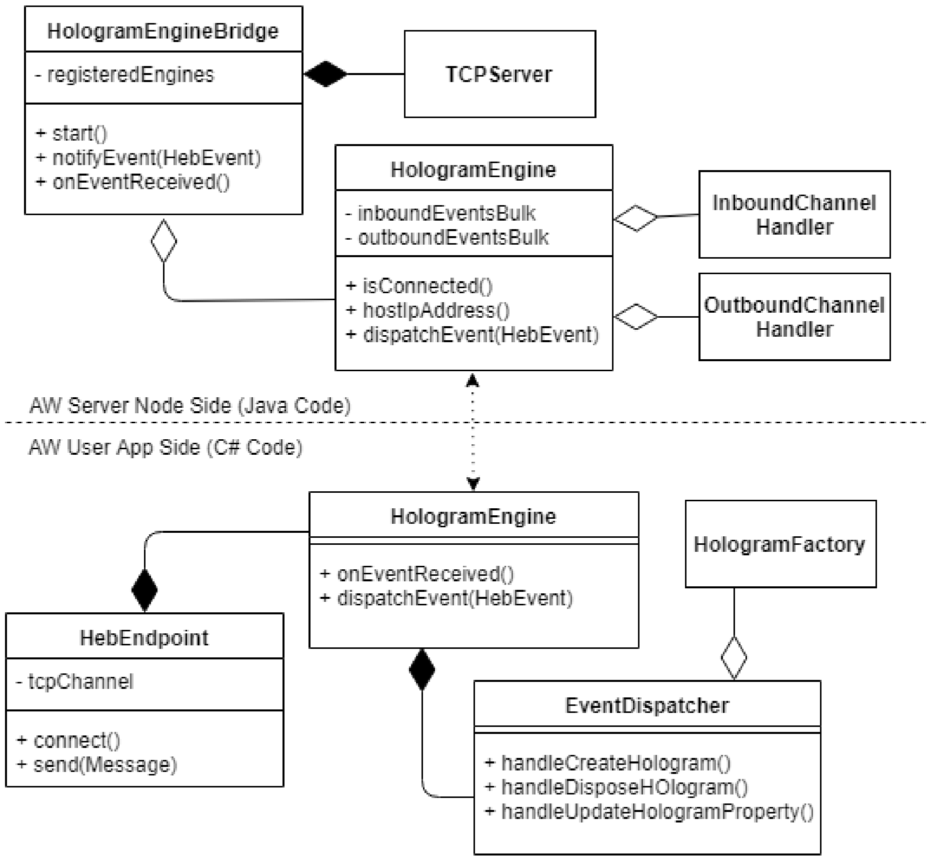


Figure 7.6: The Hologram Engine Bridge stucture model.

7.3.4 An Artifact-based API for Cognitive Agents

In MiRAgE external agents can exploit the WoAT layer offered by the AW-Runtime to interact with the AW to join the AW instance, create augmented entities, define regions, track entities and properties and so on. Currently, there are no limitations in terms of languages and paradigms to be used for code agents behaviours. An "agent" can be coded in several ways, from a simple OOP thread (e.g. using Java) to a real cognitive agent exploiting, for instance, agent-oriented languages like Jason or ASTRA. The unique constraint for the chosen language to code agents is about the availability of libraries to perform network REST request to the WoAT layer and to manage WebSockets to receive push notifications about related to possibly tracking requests.

In this dissertation we promote the usage of real agent-oriented languages to code agents for augmented worlds and, to facilitate the communication with the WoAT layer of the AW-Runtime, we have developed a library called *Mirage4Agents* based on CArtAgO artifacts to facilitate the developer task of designing agents avoiding to define an ad hoc mechanism for the web-based communication. The model proposed in Figure 7.7 reports

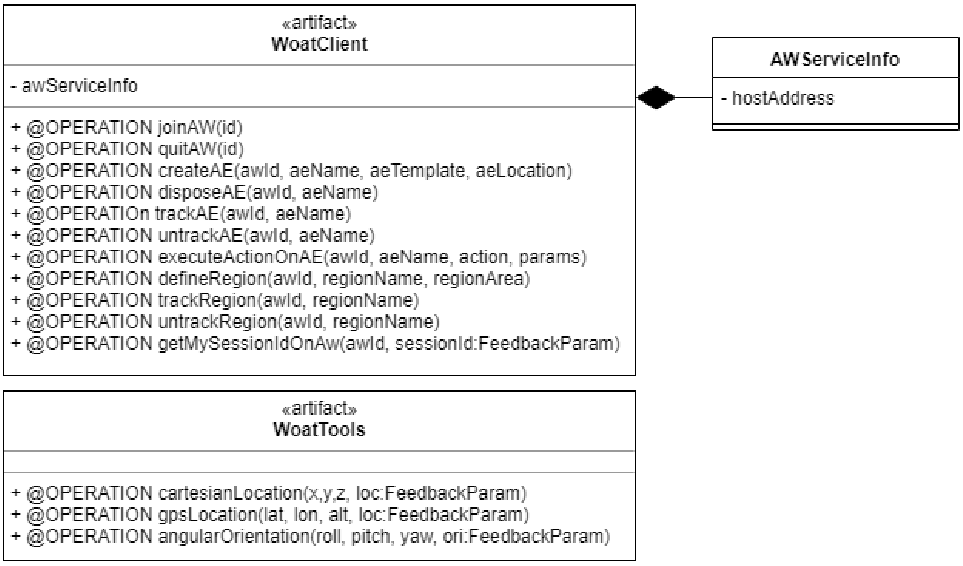


Figure 7.7: The *Mirage4Agents* lib main elements.

the main elements of this library. In particular, this library is suitable for those agents coded using agent-languages supporting the interaction with the environment based on CArtAgO. In the next future, we planned to develop also a similar library based on EIS to extends the MiRAgE support to other agent-oriented languages.

Considering the constraint of the augmented world model to request to external agents to authenticate themselves, exploiting the *sessionId* received by the AW-Runtime as a result of success for the join request, the library encapsulate also this mechanism providing to agents an artifact called `WoatClient` exposing CArtAgO operation that can be used by agents to interact with the desired running instance of an augmented world within a particular AW-Runtime. Moreover, an artifact called `WoatTools` is available, to manage augmented entities and regions ontologies, for instance, to easily create location structures and so on. As an example, in Listing 7.5 is reported a portion of the code of the `WoatClient` artifact where some implemented operations are visible.

```
public final class WoatClient extends Artifact {
  private AwService service;

  void init(final String ip) { awService = new AwService(ip); }

  @OPERATION public void joinAW(final String awId) {
    final AgentCredentials credential = /* ... */;
    service.woatAPI().joinAW(service.address(), awId, credentials, res -> {
      if(res.succeeded()) {
        final String sessionId = res.result();
```

```
        agentActiveSessions.put(awId, sessionId);
        signal("join_done", awId);
      }
    });
  }

  @OPERATION public void createAE(final String awId, final String aeName,
      final String aeTemplate, final CartesianLocation position, final
      AngularOrientation orientation) {

    final LocationProperty locProp = new LocationProperty(position);
    final ExtensionProperty extProp = new ExtensionProperty(/* ... */);
    final OrientationProperty ortProp = new OrientationProperty(orientation);

    service.woatAPI().createAE(service.address(), checkSessionId(awId), awId,
        aeName, aeTemplate, locProp, extProp, ortProp, res -> {
      if(res.succeeded()) {
        signal("ae_created", aeName);
      }
    });
  }

  @OPERATION public void trackAE(final String awId, final String aeName) {
    service.woatAPI().trackAE(awService.address(), checkSessionId(awName),
        awId, aeName, remoteObs);
    signal("ae_tracking_begin", aeName);
  }

  final RemoteObserver remoteObs = new RemoteObserver() {
    @Override
    public void onStatusChanged(final JsonObject status) {
      signal(AE_PROPERTY_UPDATE_SIGNAL, aeName, status.getString("property"),
          status.getValue("value"));
    }
  }

  /* ... */
}
```

Listing 7.5: A portion of the code of the `WoatClient` CArtAgO artifact.

Note that both operations exploit an internal layer developed using Vert.x to implement the communication with the WoAT layer of the AW-Runtime runtime encapsulating RESTful request and related content as JSON object.

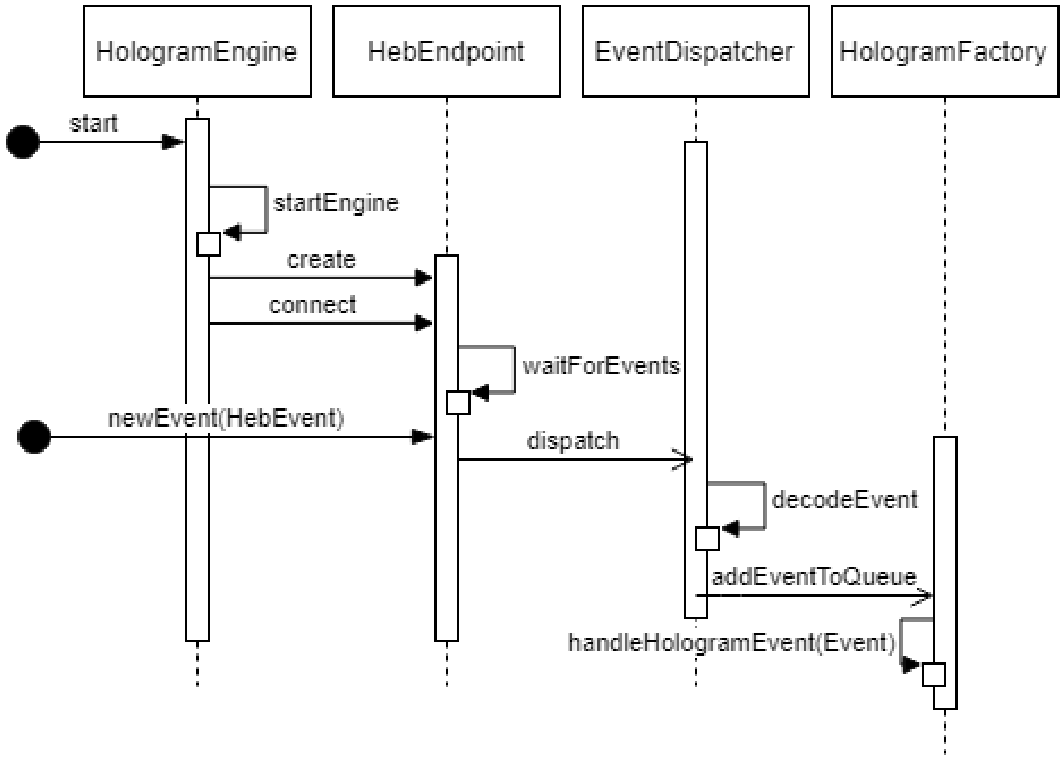


Figure 7.8: The HologramEngine interaction within the AWUserApp.

7.3.5 Building an **AWUserApp** to interact with holograms

Each AWUserApp in MiRAgE is developed as an Unity application and then deployed on the proper device through Unity deployments code conversions and mechanisms. To provide to the developer appropriate support for MiRAgE an ad-hoc Unity Package has been developed with proper tools to introduce the required components of the HologramEngine to interact with its counterpart on the AW-Runtime running on the server, where the augmented world instance lives. Considering their main high-level interactions, Figure 7.8 reports the main involved elements of the developed package.

Within the MiRAgE developed Unity package the developer will found a predefined Unity scene to build the AWUserApp composed by required components already configured (the HologramEngine, the ARCamera and many others). For all the remaining configurations – e.g., the IP address of the server node, the folder containing the Unity Prefabs representing the geometry of holograms related to each template defined for augmented entities in the AW instance – the Unity main editor has been enriched to configure the AWUserApp easily. In Figure 7.9 an empty Unity scene with MiRAgE main components is reported. Moreover, in the same figure can be observed the HologramEngine Inspector with the ad-hoc developed add-ons for the Unity editor.

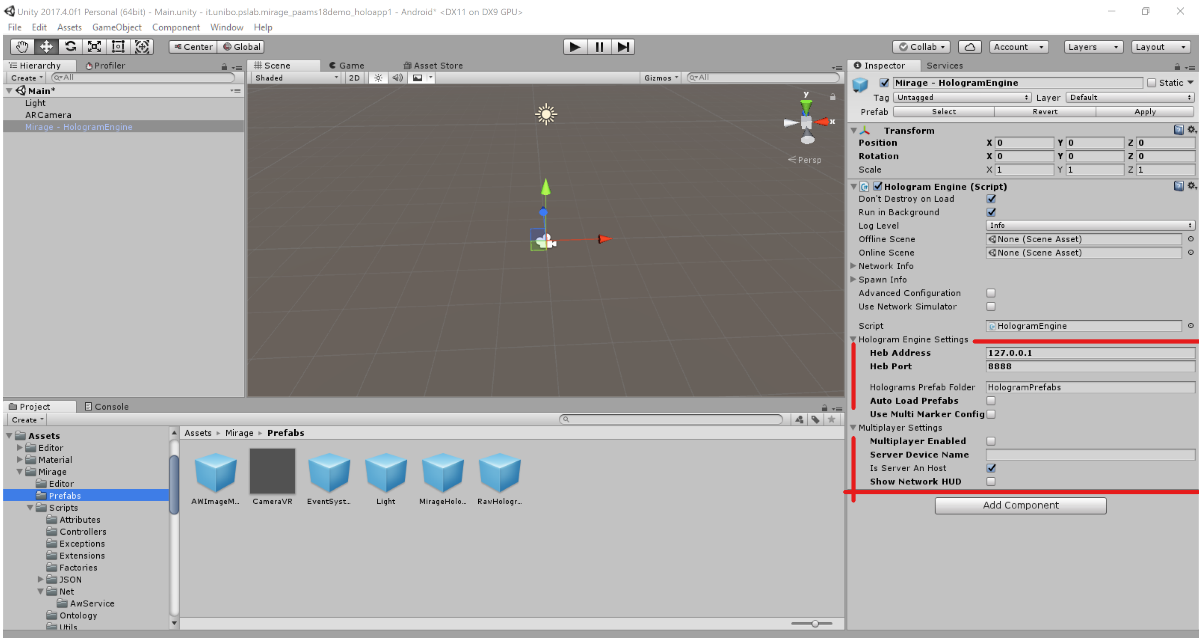


Figure 7.9: The MiRAgE Unity empty scene generate by the developed MiRAgE Unity plugin.

7.4 Developing AWs with MiRAgE

After describing MiRAgE's architectural components and main elements of its developed prototype, this section aims to offer the reader a list of the methodological steps that ought to be undertaken for developing an AW's instance and how these steps are related to prototype elements.

Firstly, it is important remembering that each AW instance built exploiting MiRAgE is intrinsically a distributed software system allowing for a multi-user experience by construction: this means that the programmer does not have to be worried of developing this level/feature for the system. Each instance of an AW is composed at least by three different subsystems to be developed and executed, in particular:

- the "*aw-core*", containing the AW-Runtime and the running instance of the AW along with the defined template for each instance of augmented entities;
- the "*holograms-app*", defining the representation/geometry in Mixed Reality of each involved hologram to be deployed on each user device;
- the "*agents-app*", defining the behaviour of the MAS and agents involved in the AW.

Using MiRAgE the task to develop and run the "*aw-core*" is limited to defining a Java application responsible for configuring and running the AW instance through the APIs provided by the framework (in particular, whose provided by the `Environment` library). This application has to include the templates representing each augmented entity instance that agents could create and run. In MiRAgE an AE template is defined in OOP-style, defining proper classes ad-hoc annotated. Listing 7.6 reports an example of a trivial AE template.

```
@HOLOGRAM(geometry = "CubeHologram")
public class Cube extends AE {

  @PROPERTY private int side;
  @PROPERTY private Color color = "white";

  @ACTION void resize(Double newSideSize) {
    customProperty("side", newSideSize);
  }

  @ACTION public void rotate(Double degrees) {
    AngularOrientation orientation = (AngularOrientation) orientation();

    orientation(new AngularOrientation(
      orientation.roll(),
      (orientation.pitch() + degrees) % 360,
      orientation.yaw()));
  }
}
```

Listing 7.6: An example of an AE template in MiRAgE

Each template uses ad hoc annotations to define AE's properties and actions. Then, at the runtime, the awr will be responsible for creating each instance of an AE considering each invocation of the `createAE()` API called by agents defined in the "*agents-app*". For every AE creation, the involved agent is responsible for defining – in a JSON coded message – the initial configuration of the AE, mainly to provide a value for default properties, e.g. location, orientation, and so on. So, starting from the specific template, the AW-Runtime creates the AE with custom properties and actions. In the case of the template also defines a geometry for the AE, the AW-Runtime also informs the HologramEngine exploiting the Hologram Engine Bridge in order to create and show the related hologram.

The "*agents-app*" is a (set of) MAS running all agents that in a particular instant ahead of time will join the AW and will start to act and observe the evolving state of entities. Each agent interacts with the AW-Runtime exploiting the *Mirage4Agents* library, currently developed as a set of artifacts in CArtAgO. Despite this, the usage of this library is not mandatory. Thanks to the WoAT layer of MiRAgE, each proactive software component – not necessarily a cognitive BDI agent – able to perform REST-oriented web requests can join and interact with a running instance of an AW.

Finally, at the HologramEngine side – the "*holograms-app*" – after an initial configuration of the environment provided by the Unity MiRAgE provided package, the focus of the developer has to be oriented in defining the geometries of holograms and, eventually, their responses to custom properties updates of the attached augmented entities: this can be done in MiRAgE defining ad-hoc controllers (C# scripts) coded directly in Unity. The complete process for building and deploy each instance of HologramEngines for each AWUserApp is managed autonomously by Unity. The HologramEngine manage autonomously movements of holograms in the Mixed Reality scene and inputs from users on them.

As a final remark, for a more exhaustive description of how to build an AW using MiRAgE we redirect the reader to Section 8.1 of the next Chapter, where a non trivial case study will be described and will also be used for a first evaluation of the framework and the MiRAgE approach.

7.5 MiRAgE Extensions

In this section we are going to describe some extension developed contextually to the MiRAgE platform, to integrate essential aspects making MiRAgE more useful and usable for developing first cases studies considering the augmented world's vision. Other extensions are currently under evaluation and will be developed soon in the next future.

7.5.1 A direct support for the Meta2 MR visor

In section 2.4 we have presented the Meta2 visor, discussing its features and affirming that nowadays it one of the best device for offering an appropriate mixed reality experience to users. Considering this and considering the availability in our labs of two of these devices, we have developed an extension for MiRAgE to directly integrate this visor in an environment where users can enter and see holograms both using, e.g., tablets/smartphones and also the Meta2 visor simultaneously sharing the same experience.

Developing applications for Meta2 require basically the realization of an application based on Unity with the Vuforia plug-in. Thus, any MiRAgE AWUserApp – running an instance of the HologramEngine – can technically be easily executed on the Meta2 visor. Despite this, the real issue in using the Meta2 visor is related to the registration of holograms. In fact, at it starting phase, Meta2 exploits its sensors to scan the real environments for building a digital map of it, assuming the center of the digital world coupled with the position of the user wearing the visor and oriented according to its

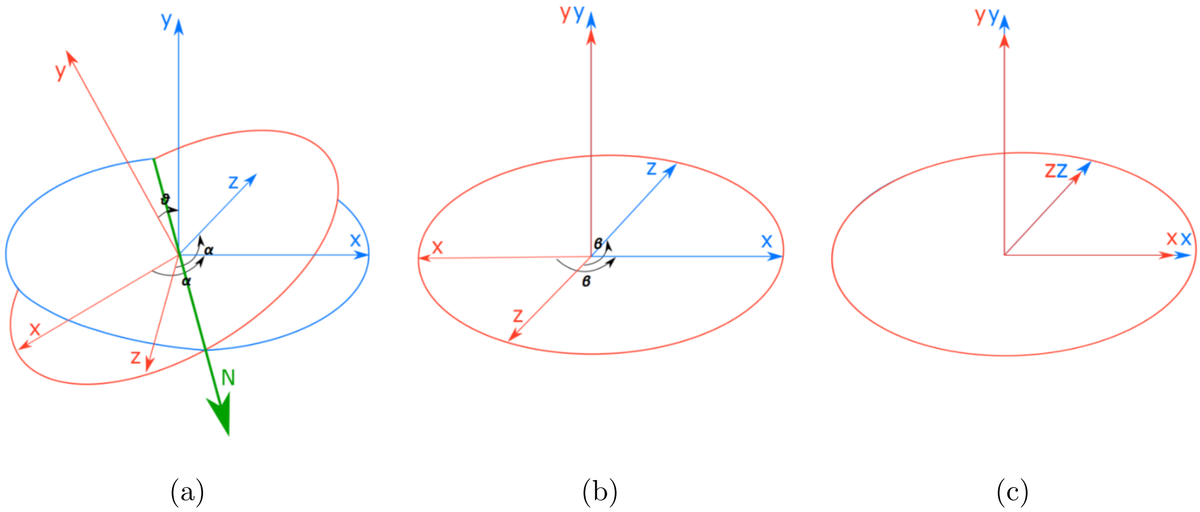


Figure 7.10: A representation of performed algorithm steps to align the two reference systems. (a) The two reference systems unaligned. (b) The two reference systems aligned on the y-axis using the registration procedure. (c) The two reference systems completely aligned after the application of the rotation algorithm.

current gaze. This mechanism contrasts with the AW model, where the reference system is unique for the AW and is shared among humans and agents. So, to allow the usage of the Meta2 visor as a medium for a user to enter in the augmented worlds, a procedure to real-time aligning the visor built-in reference systems with the one used within the runtime AW instance is required.

Entering into details, the HologramEngine component of MiRAgE has been extended to consider a bidirectional transformation of holograms coordinates to be applied in case of usage of the Meta2 visor — see the conceptual representation of this procedure in Figure 7.10. In particular, a HologramEngine running on a Meta2 device applies this transformation

1. every time that a new hologram has to be rendered or updated, transforming locally the holograms coordinates expressed in terms of the AW instance reference system to local coordinates used by the visor;

and

2. every time that an input action on a hologram, e.g. the user grabs an edge of a hologram, has to be notified to the corresponding augmented entity in the AW, transforming local coordinates into AW global coordinates.

To register the transformation matrix to be used to convert the coordinates, the HologramEngine requires that the user performs a “registration step” before entering in the AW.

This registration step consists in looking to a particular marker – possibly the same used to keep the AW origin for others mixed reality techniques – to identify current position and orientation to be used to build the transformation matrix.

This approach suffers from some problems, to be resolved in future work. For instance, because visor cannot be assumed to be stable with a particular orientation – since the user wears it – this aspect introduces some deltas in the registration process that causes a little degree of non-alignment between the two, the local and the global, reference systems. Nevertheless, this approach could be used not only for integrating the Meta2 visor in MiRAgE but also to consider all device supporting SLAM techniques and that have sensors for a depth-scan of the real environment such as, e.g., Microsoft HoloLens.

7.5.2 Tracking non-computational physical things

Although it is still in an early stage of development, another extension developed for MiRAgE is about the idea to allow agents to identify and "acts on" non-computational physical things. To clarify, imagine to have a real bottle on a real table and a digital holographic character moved by an agent on the table with the purpose to avoid to collide with the real bottle. The AW conceptual model allows doing this merely assuming that the real bottle is modelled as augmented entities, with the current position of the real object and its extension in the 3D real space. Thus, shaping the bottle in the digital space allows the agent to quickly define a behaviour to move on the table, skipping the physical things.

The point here is how to introduce run-time in the digital space non-computational things – i.e., without computational capabilities – and keeping track of related changes, i.e., in terms of position in the real world and extension. To do this, the idea is to exploit the mesh of the environment generated by mixed reality visors – such as Meta2 or HoloLens – using SLAM registration techniques. Once identified the physical thing of interest, using the visor generated physical environment mesh, an ad hoc developed tool collateral to MiRAgE can be used to capture point of interest from the mesh and define an invisible *bounding box* for the interested physical thing. For instance, in the used example, a cylinder can be used to define a bounding box for the physical bottle.

The bounding box can be used, thus, as the geometry of the "hologram" attached to the augmented entity representing the physical thing (see Figure 7.11). Considering this workaround, also non-computational physical things can be modelled in the digital environment and tracked by agents acting in the AW-Runtime. For instance, we can detect when a user grabs a physical thing, observing when the user grabs the invisible bounding box modelling the geometry of the object. Vice versa, exploiting physic's laws implemented within each HologramEngine, a physical thing can be used to hit a hologram, moving it.

This approach only partially solves the issue. In fact, the major constraint is related to the fact that each movement in the real world performed on a physical thing coupled

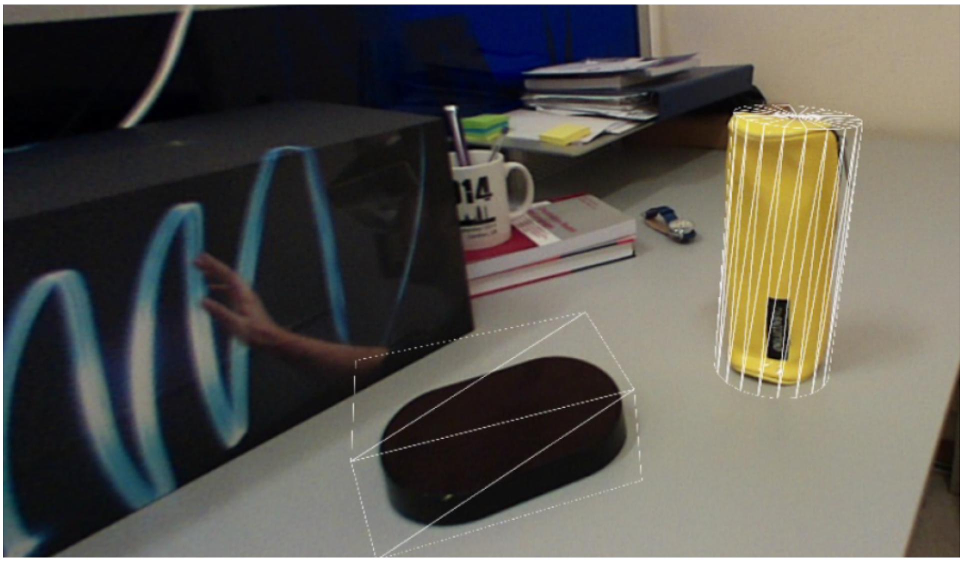


Figure 7.11: A bounding box rendered over a physical cylindric object.

with a proper bounding box should be done only when the portion of the real environment where the physical thing is placed, is under tracking by a SLAM visor. If not, the spatial coupling between the position of the real object and its related bounding box would inevitably be lost. Future works will be devoted to refining this approach, integrating advanced computer vision techniques to identify objects within the real-time scene. In any case, despite this current relevant limitation, this MiRAgE extension is promising to expand potentialities and feature offered to augmented worlds developers.

Chapter 8

MiRAgE Evaluation

This chapter aims to propose a first evaluation of the MiRAgE platform described in the previous Chapter, discussing its essential aspects and proposing a first case study where MiRAgE has been used for developing pervasive mixed reality smart environments. Currently, the MiRAgE platform exists in the form of a prototype, and for its nature exposes many aspects which can be improved that will be taken into account in future works. Nevertheless, the current state of the infrastructure allowed us to effectively use it to develop some case studies, one of them already discussed as a step-by-step example in Section 8.1, to test and improve the prototype as is to refine the augmented world conceptual model. In this chapter we will evaluate the infrastructure in terms of performances, effectiveness and scalability, leaving a complete evaluation to the overall discussion addressed in the next chapter, that will also involve consideration on the augmented world's model and more.

8.1 The "*Augmented Museum*" case study

The aim of this section is to give a taste of AWs design and development, by considering a simplified version of a real-world case study – generally called *augmented museum* – that is part of a cultural heritage project within the context of maritime archaeology, developed in the context of the European *Society* project in cooperation with the "*Museo della Regina*", an Italian museum in Cattolica (RN). Also, the case study allows for a qualitative evaluation of the applicability of the AW conceptual model and the infrastructure for designing a real instance of an AW.

Generally speaking, the objective of the project was to investigate the design of augmented worlds layered upon an existing physical environment, a room in a museum to enrich the experience of the visitors. A collection of photos of the real-world running demonstration is proposed in Figure 8.1. In this case, the AW is composed by an ancient Roman boat autonomously navigating over a virtual sea place among a set of real ancient

Figure 8.1: Some photos of the main part of the Augmented Museum installation: a visitor watching – using a tablet (*top-left*) and wearing smartglasses (*bottom-left*) – the boat moving in the virtual sea placed within a set of ancient amphorae arranged in a circle.

amphorae, arranged in a circle. If/when the boat hits a physical amphora, it goes back and changes direction. If/when the boat enters into a specific region ("red-zone") of the virtual sea, then a physical small lighthouse switches on its light, which are switched off as soon as the boat lefts the zone. Multiple human visitors (users) can enter the AW, sharing the experience and interact with the boat. Visitors can generate some wind, affecting the trajectory of the boat. The coupling between the AW and physical world is based on markers placed into the environment, at the center of the stage.

Even if quite classic and straightforward, this example involves some of the primary key points about AW: holograms featuring an autonomous behaviour interacting with the physical world, bi-directional augmentation, multi-user system. Figure 8.2 shows an abstract representation of the AW setting, along with the main architectural elements (agents and augmented entities) involved in the given example.

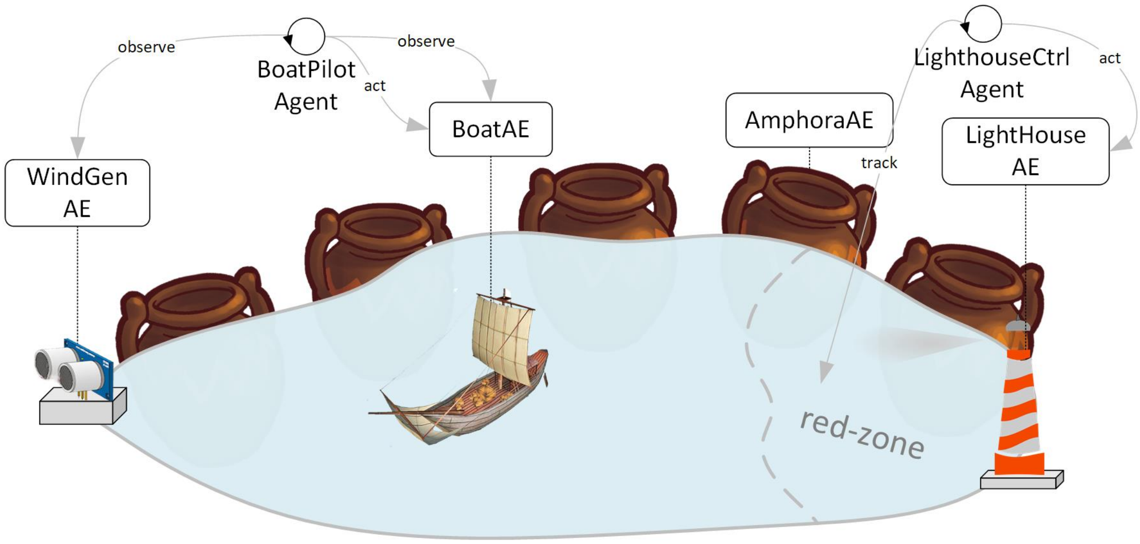


Figure 8.2: Augmented museum example: AW design and involved elements.

8.1.1 Environment set-up

Exploiting dedicated MiRAgE libraries, the AW environment for this demo has been setted-up with a deployment on a single node according to specifications defined within the `environment-config.aw` file (see `Environment.deployRuntime()` function).

After having deployed the AW-Runtime, a simple instance of an AW (identified by the id `augmented-museum-demo-aw`) has been started within the AW-Runtime running on the selected node (see `Environment.startNewAugmentedWorldInstance()` function that requires to specify the type of environment to be used, the selected reference system – in this case `CARTESIAN` due to the indoor nature of the demo – and the extension of the region covered by the AW).

Listing 8.1 reports the simple code used for these two operations. Note that in such a very immediate way, an instance of an AW can be created and deployed in MiRAgE.

```
public class AugmentedMuseum {
   public static void main(String[] args) {
      Environment.deployRuntime("aw-settings/environment-config.aw");
      Environment.startNewAugmentedWorldInstance("museum-demo-aw",
             AwEnvironment.AR, AwReferenceSystem.DEFAULT,
             AwReferenceSystemType.CARTESIAN, ExtensionType.BASIC, /*...*/);
   }
}
```

Listing 8.1: The AW instance setup code.

8.1.2 AW entities initialization

When the AW is running, in this demo we decided to delegate to an autonomous software agent coded in Jason – called InitializerAgent – the task to create all involved augmented entities. To accomplish this task, the agent joins to the AW and exploits the developed *Mirage4Agents* library (`woatClient` and `woatTools` artifacts in particular) to create augmented entities and regions for the involved application logic elements. Such artifacts provide access to agents APIs to communicate with the WoAT layer reachable exploiting the network on the node where the AW-Runtime is running (in this case, the IP address of this node is related to localhost). Listings 8.2 reports a snippet of the relevant parts of this agent.

```
serverIpAddress("127.0.0.1").

!start.

+!start : serverIpAddress(Ip)
  <-  makeArtifact("woatTools", "mirage4agents.WoatTools", [], _);
      makeArtifact("woatClient", "mirage4agents.WoatClient", [Ip], Lib);
      focus(Lib);
      joinAW("museum-demo-aw").

+join_done("museum-demo-aw")
   <- sessionIdOnAw("museum-demo-aw", SessionId);
      .concat("Agent has joined the aw! Session ID = ", SessionId, Msg);
      println(Msg);
      !initAugmentedMuseumAwEntities.

+!initAugmentedMuseumAwEntities
  <- !createLighthouse("lighthouse");
     !createBoat("boat");
     !createRedZoneRegion("red-zone").

+!createBoat(Name)
   <- cartesianLocation(0, 0.005, 0, Loc);
      angularOrientation(-90, 0, 0, Ori);
      createAE("museum-demo-aw", Name, "Boat", Loc, Ori).

+!createLighthouse(Name)
   <- cartesianLocation(0, 0.095, 2.5, Loc);
      angularOrientation(0, 0, 0, Ori);
      createAE("museum-demo-aw", Name, "Lighthouse", Loc, Ori).
```

```
+!createRedZoneRegion(Name)
  <- cartesianLocation(0, 0, 2, Position);
     circularCartesianArea(Position, 0.8, Area);
     defineRegion("museum-demo-aw", Name, Area).
```

Listing 8.2: The InitializerAgent code.

Regarding the augmented entities, the agent creates the BoatAE and the LighthouseAE assuming that in the AW-Runtime would be available the required templates defining the structure in terms of properties and actions of these augmented entities. A snippet of the Java classes, properly annotated, representing those templates are reported in Listing 8.3 and Listing 8.4 respectively.

```
@HOLOGRAM(geometry = "Boat")
public class Boat extends AE {

   @PROPERTY private double speedVal;
   @PROPERTY private Vector2D speedVersor;
   @PROPERTY private int windForce;

   @ACTION void setSpeed(Vector2D versor, double val) {
      customProperty("speedVal", val);
      customProperty("speedVersor", versor);
   }

   @ACTION void setWindForce(double windForce) {
      customProperty("windForce", windForce);
   }
}
```

Listing 8.3: The BoatAE template definition.

```
@HOLOGRAM(geometry = "Lighthouse")
public class Lighthouse extends AE {
   @PROPERTY private String lightStatus = "off";
}
```

Listing 8.4: The LighthouseAE template definition.

In these templates definitions, note the usage of @HOLOGRAM, @PROPERTY and @ACTION annotations to identify, respectively, the Unity prefab defining the geometry of the hologram, the observable properties and the actions of the entity. Other involved augmented entities involved templates are not reported due to their simplicity.

8.1.3 Defining the application logic

At the design level, the boat pilot agent – called BoatPilot – encapsulates the autonomous behaviour of the boat, while the body and hologram of the boat are represented by the BoatAE augmented entity, previously defined. In Listing 8.5 is reported a snippet of the code of the BoatPilot agent, implemented in Jason.

```
!init.

+!init
  <- joinAW("museum-demo-aw").

+ae_created("boat")
  <- trackAE("museum-demo-aw", "boat");
     !navigate(30).

+ae_created("wind")
  <- trackAE("museum-demo-aw", "wind").

+!navigate(Speed)
  <- getRandomOrientation(Versor);
     executeActionOnAE("museum-demo-aw", "boat", "setSpeed", Versor, Speed);
     .wait(5000);
     !navigate(Speed).

@manage_border_collision[atomic]
+borderReached
  <- !divertBoatRoute(180).

+ae_property_update("wind", "force", V)
  <- executeActionOnAE("museum-demo-aw", "boat", "setWindForce", V).
```

Listing 8.5: The BoatPilot agent code.

The goal of the agent is to continuously move the boat according to a random trajectory and react to relevant events. The agent tracks (observes) BoatAE so as to react to a collision event. The collision event is generated when the BoatAE collides with any AmphoraAE augmented entities, which are augmented entities modelling/coupled to the real amphorae. The agent tracks also WindGenAE, to react to changes into wind force and acts on BoatAE accordingly. This WindGenAE augmented entity in this demo is coupled to a physical Thing, embedding an Arduino single-board micro-controller with a proximity sensor, for detecting the distance of user's hands, simulating the wind. In a WoT-style, this thing can be accessed using a REST web API, used in the WindGenAE implementation. To that purpose, the embedded system uses a Bluetooth connection to a gateway, based on Raspberry PI 3, acting as a bridge to the network.

```
!init.

+!init
  <- joinAW("museum-demo-aw").

+region_defined("red-zone")
  <-   trackRegion("museum-demo-aw","red-zone").

+region_entities_update(X, Entities)
  <-  !checkBoatInRedZone(Entities, "boat").

+boat_in_red_zone("yes")
  <-    executeActionOnAE("museum-demo-aw", "lighthouse", "switchOn").

+boat_in_red_zone("no")
  <-    executeActionOnAE("museum-demo-aw", "lighthouse", "switchOff").

+!checkBoatInRedZone([], E)
  <- -+boat_in_red_zone("no").

+!checkBoatInRedZone([H|T], H)
  <- -+boat_in_red_zone("yes").

+!checkBoatInRedZone([H|T], E)
  <- !checkBoatInRedZone(T, E).
```

Listing 8.6: The LightHouseControllerAgent code.

The MAS includes also the LightHouseControllerAgent (see Listing 8.6), which is tracking the region corresponding to the red-zone and switches on/off the light by acting on the LighthouseAE augmented entity. Also, this augmented entity is coupled to a physical thing, based on Arduino. This agent reacts to the entrance/exit of any augmented entity (the boat in this case) and performs switchOn/switchOff action on the augmented entity.

From the HologramEngine side, each user involved in the demo is supposed to be equipped with an Android device (either a tablet or smartglasses) to see and interact with the AW. In Figure 8.3 is proposed the Unity IDE where the development of the AWUserApp has been carried out. In particular, note the presence in the scene of the two main components of the MiRAgE plug-in for Unity: its camera and its HologramEngine. Moreover, within assets, the two prefabs representing geometries for holograms related to augmented entities templates has been adequately realised.

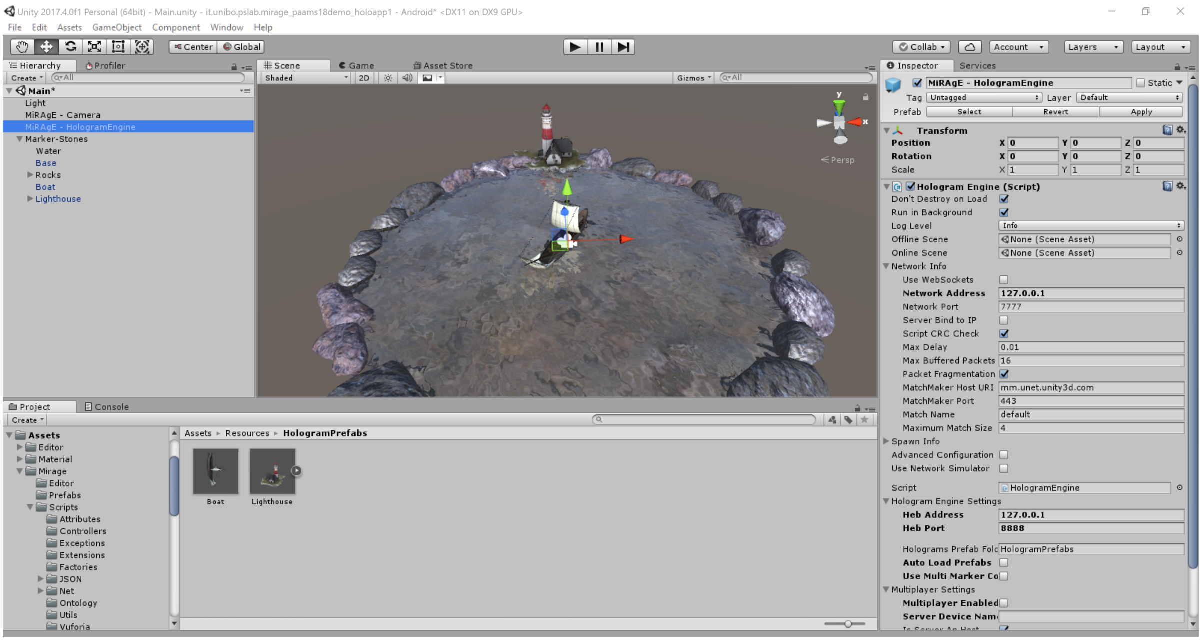


Figure 8.3: The HologramEngine Application on Unity.

About the code that a developer has to produce at the HologramEngine side, in the case of this demo, the only hologram that needs to change it representation according to a change to the value of a custom property of its augmented entity is the lighthouse hologram, that need to turn on/off its light. To deal with this requirement, to the Lighthouse prefab has been attached a proper controller reported in Listing 8.7 that is responsible for reacting to custom properties updates.

```
public class LighthouseController : HologramController
{
  public GameObject lightObject;
  private const string LIGHTSTATUS_PROPERTY = "lightStatus";

  public override void onCustomPropertyUpdate(string property, object value)
  {
    switch (property)
    {
       case LIGHTSTATUS_PROPERTY:
         manageLightStatusPropertyUpdates(value);
         break;

       default: throw new NotImplementedException();
    }
  }
```

```
  private void manageLightStatusPropertyUpdates(object status)
  {
    if (status.ToString().Equals("on"))
      lightObject.SetActive(true);
    else if (status.ToString().Equals("off"))
      lightObject.SetActive(false);
  }
}
```

Listing 8.7: The Lighthouse Hologram Controller.

That's all! The reported code is the only requested code to be written to run the demo in a completely distributed and multi-user environment designed following augmented worlds vision and model. The infrastructure support itself the dynamic join of agents and especially users, avoiding to the logic application developer to be worried about this. Moreover, as previously defined, MiRAgE also support the coupling with the physical entities and things by construction. The focus of the designer/developer should be only devoted to design involved elements such as agents and augmented entities.

8.1.4 Considerations

The case study presented in this section is undoubtedly a quite simple case study. Despite this, it is an example that involves all the crucial aspects of AWs: it is composed by several AEs with proactive behaviour managed by cognitive agents, involves interaction with physical computational things, proposes a visualisation of multiple holograms in a multi-user context, involves non-computational physical entities, and so on. For this reason, it is undoubtedly a relevant case study that makes it possible to have a first evaluation on the applicability of the MiRAgE framework and the level of abstraction offered to the programmer. Although a direct and functional comparison with a comparable software system developed without the aid of MiRAgE has not been done, author's experience allows us to affirm that, at least, MiRAgE allows to significantly reduce the time required to develop and run an AW, avoiding needing to start developing by scratch.

As all designer and developers of smart Mixed Reality systems well known, among mainstream technologies there aren't robust tools to efficiently manage context like AWs. In particular, multi-user contexts where cooperatively and in real-time users can see multiple holograms considering their perspective, especially in the case that holograms are not just static elements but full-fledged computational entities with an evolving state over time. Conversely, MiRAgE address all of this issues by construction, allowing the developer to focus only on the design of the application logic of the AW, avoiding to deal with lower-level issues like set-up a shared real-time environment and so on.

Another relevant consideration emerging from this case study is about the adequateness of the designed abstractions. In particular, the ontology for AEs, regions, and other model elements seem to cover well all essential needs that a programmer wants when has to set up such a mixed reality software system.

Finally, this case study allowed also for a preliminary evaluation of provided APIs for agents. Emerged that despite the provided APIs offered all required functionalities with a very high level of abstraction, a more specific subset of APIs could be useful, especially when the behaviour of agents for articulated augmented entities become more complicated than in trivial examples.

8.2 Functional Evaluation

Evaluating appropriately and significantly a heterogeneous infrastructure like MiRAgE, unfortunately, is not an easy task to accomplish. First of all, it involves several technologies, and this fact introduces a certain degree of complexity in the evaluation process. Also, the nature of involved systems – pervasive computing systems and mixed reality ones that are not easy to evaluate and test by themselves – make the evaluation process a hard process.

However, from a functional point of view, developed case studies and ad hoc developed test applications highlighted how MiRAgE is an infrastructure able to incredibly reduce the existing gap from available technologies for developing mixed reality applications and requirements of smart environments here called augmented worlds.

Developing an AW using MiRAgE lets the programmer to think to the final distributed system as a more straightforward concentrate application. Each instance of an AW could be considered by default as a distributed system: MiRAgE allows to build the AWUserApp running on each user device and also the instance of the AW running on the AW-Runtime can be easily distributed over more network nodes. Moreover, even the cooperation among humans and agents can be "forgotten" by the developer: is in charge of MiRAgE to produce a cooperative and collaborative environment where the communication among users/agents is managed as a first class element. Exploiting observation of augmented entities properties and tracking feature, agents can reason and act over the environment to perform their tasks and augment the physical world. Agents do not be designed to be necessary informed of the physical world, because the digital layer coupled with the physical one provides to them the right level of abstraction to reason and act upon.

Apart from existing (but not functionally relevant) bugs and not yet implemented features, MiRAgE for its nature could be considered a candidate to become a reference platform for building general-purpose augmented worlds, especially for those smart environment where bidirectional augmentation between physical and digital layers is the very starting point. Besides, MiRAgE could be a solid infrastructure especially for those developers coming from the field of agents and MAS, wanting to bring their systems in the physical world, making agents able to interact with the physical world and in particular humans easily and vice versa.

Finally, summarizing MiRAgE's strong points, among others, four are the main critical functional aspects:

1. providing to a developer all main abstractions needed to design the application logic of an instance of an augmented world;

2. offering complete support for mixed reality interaction avoiding to directly take into account related issues like registration, multi-user real-time interaction, etc.;

3. for programmers coming from agents and MASs community, providing an infrastructure that allows them to experiment smoothly, e.g., applications of artificial (ambient) intelligence in the real environment with a platform useful both for simulations and for real interaction with humans and things;

4. for programmers coming from Mixed Reality community, offering them the ability to define complex software systems with a high degree of pro-activeness exploiting adequate abstraction like cognitive agents for building holograms behaviours, even in this case both in simulation or in real case studies and systems.

Surely, MiRAgE has also some limitations and critical points to be kept in mind, e.g., in terms of performances or if we consider latencies introduced by the full distributed approach and the fact that the model requires to separate the real-time state of the augmented world by its, possibly, holographic representation for humans. However, we have to avoid forgetting that MiRAgE has been built to bring together Mixed Reality with Pervasive computing and MAS. So, from a functional point of view, it is important the high level of abstraction provided in building such integrated complex software systems and not a comparison with each single areas.

8.3 Performances and Effectiveness

Beside the functional evaluation, several considerations about performances offered by MiRAgE have been carried out in order to analyse the developed prototype of the infrastructure and to identify limitations of the approach and which parts and components to improve in future works. Generally speaking, from an engineering point of view, some main critical point of the MiRAgE framework could be identified in:

- the degree of real-time and effectiveness related to the interaction of the external agents with the AW's entities through the WoAT layer;

- the issues related to the propagation of properties changes from augmented entities to each HologramEngine, considering the impact on the human users, considering the cooperative nature of augmented worlds;

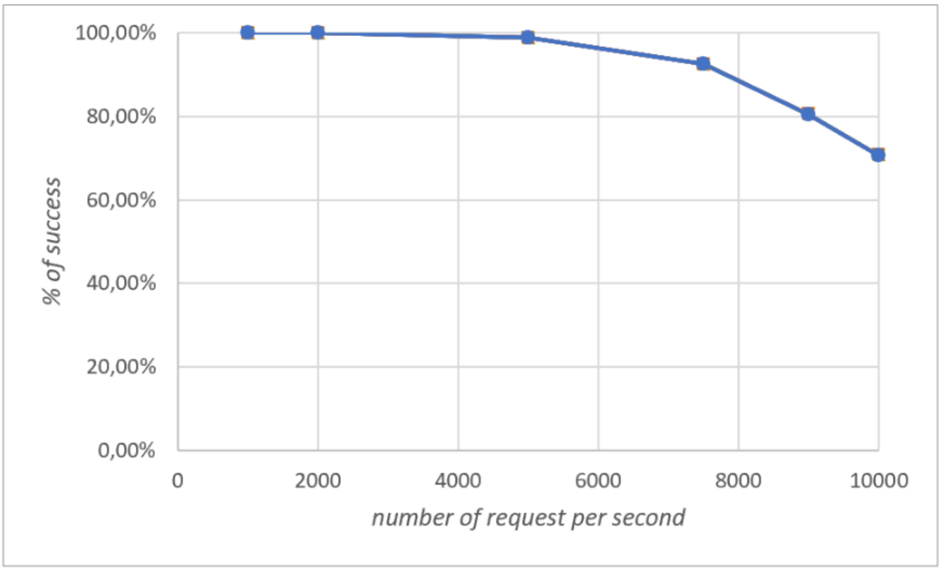


Figure 8.4: A simulation about the percentage of success of concurrent request sent to the MiRAgE WoAT APIs.

- the degree of scalability of the adopted approach.

Following paragraphs will investigate this critical point in terms of performances of the currently developed prototype of MiRAgE and highlighting the areas of the prototype where to carry out a more in-depth investigation and a refactoring of the developed specific solution.

Performances related to the WoAT layer

About the first reported observation, the issue arises because the WoAT layer – as in the WoT – use the Web as the communication medium. However nowadays it represents the standard for almost all novel software systems and promises wider interoperability respect to other approaches, the intrinsic asynchronous communication protocol has to be taken in account in evaluation performances of a software system that should be considered real-time as much as possible. In a web-based communication protocol, HTTP requests can be affected by time-outs and the responsibility to in performing again the request in case of errors is in charge of the client, agents and external entities in the case of augmented worlds. All of this aside, the developed platform has been stressed in particular to identify the number of HTTP requests that could introduce an unacceptable degree of performances within the WoAT layer. In particular, we have observed that the current state of the developed prototype allows us to efficiently satisfy up to about 2000 requests per second sent to the WoAT layer by a set of heterogeneous agents. Exceeding

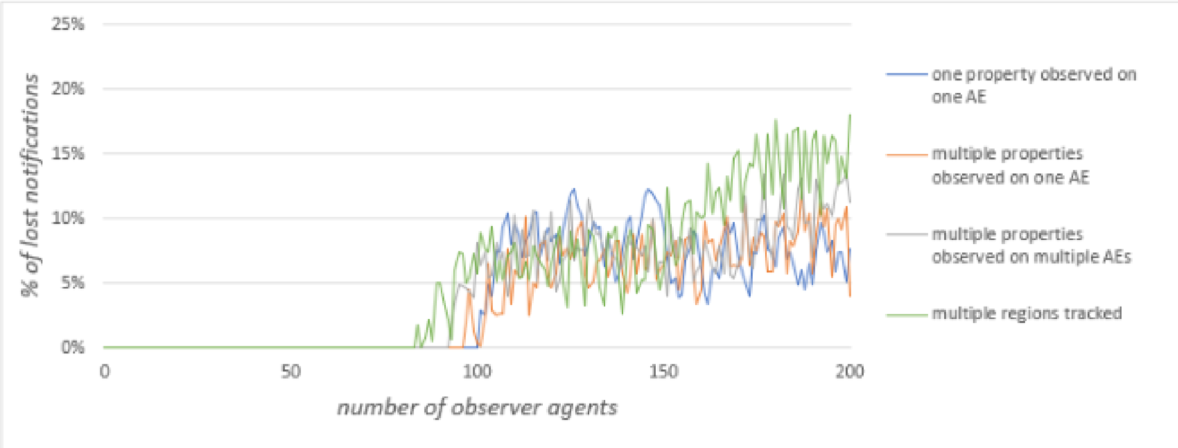


Figure 8.5: A simulation about the percentage of lost notifications for agents involved in the AW.

the number of 2000 requests per second, some HTTP requests could fail. More in detail, a success rate of 98.8% has estimated by carrying out a test with 5000 simultaneous requests. This percentage drops to 70.74%, instead, for a total number of requests equal to 10000 (see graph reported in Figure 8.4).

Finally, about the tracking functionalities (i.e. for agents, the ability to receive updates through notification when something tracked – an augmented entity or a related property, a region and so on) we have stressed the prototype and we have observed that current implementation is effective with the limit of 100 observers/trackers active for each entity. Exceeding this limit, there could be lost updates in notifying changes in entities/regions/properties under observation (see graph reported in Figure 8.5).

Future works in the development of the MiRAgE platform will be devoted to analysing this performance issue better, improving the framework to become more robust from the WoAT side. Despite this, however, the so far obtained result can be considered promising and suggest that the design road followed for MiRAgE could be the right path to continue in the direction of developing a reference platform allowing for the modelling and the engineering of augmented worlds.

Performances related to the HologramEngine

Proposing an effective experience with holograms in the Mixed Reality context is a hard task to accomplish especially when the mixed reality experience must be cooperative and shared among human users. Currently, the HologramEngine component of MiRAgE exploits Unity and the Vuforia. Such technologies by their nature help to have an effective mixed reality experience for users, due to their degree of precision in managing mixed reality content using appropriate algorithms and computer vision techniques.

Nevertheless, in software-based smart environments like augmented worlds the issue of effectiveness of the mixed reality experience is affected by the fact that the state of the involved entities is managed outside each HologramEngine and each running AWUserApps has to receive in real-time changes to holograms status (such as location, orientation, and so on).

This represents a challenge for the infrastructure, that at the current level of development of the prototype is satisfied only if not so much UserApp are simultaneously connected to the AW-Runtime, currently in the order of 10-15 users per each instance of the augmented world (however, the theoretical limit of the infrastructure is indefinite). Unfortunately, this aspect is also not simple to be numerically evaluated. To perform a significant evaluation each instance of HologramEngine should be run in a real physical device, and the availability of real device for this dissertation is currently limited.

Despite this, future work will investigate how to improve this MiRAgE limitation. Anyhow, this restriction related to the number of involved human users does not reduce the overall value of the framework in terms of research advances and for further explorations in using MiRAgE for developing first augmented worlds real case studies and applications.

Issues about scalability of the AW-Runtime

The AW-Runtime component of MiRAgE has been designed to be scalable in any direction. Currently has been tested on a single server configuration but technically it could be easily extended to be executed on distributed configuration. The AW-Runtime in execution on a particular server can support both that the computation about each augmented entities is performed locally or on a remote machine, also considering a cloud-based deployment. More developing work has to be done to complete this MiRAgE feature.

Admittedly, the scalability of the infrastructure is a critical element to be thoroughly investigated. Nevertheless, the current configuration is not a bottleneck for developing first case studies to experiment with the applicability of the augmented world model.

Part IV

Conclusions and Future Directions

Chapter 9

Discussion

After presenting the vision concerning both a proposal for a meta-model to make uniform the design of pervasive mixed reality software systems – referred here as Augmented Worlds – and a concrete platform to engineering them – the MiRAgE framework – in this chapter we want to discuss the proposed approach in terms of strengths and limitations. Moreover, we want to propose a research agenda for future works and further in-depth analysis.

9.1 Overall Evaluation

The Augmented World vision and approach proposed in this dissertation is about conceiving a new generation of smart environments as pervasive mixed reality software systems where both humans and software proactive agents cooperate and collaborate through a hybrid medium built upon the concept of augmented entities. Opportunities given both by Mixed Reality and by Pervasive Computer techniques allow us to put together the physical world with the digital one to conceive novel ways to enrich our reality with innovative computational capabilities and features.

As reported in the first chapter of this dissertation, the time is ripe for mixing augmentation technologies – considering their broadest meaning – to produce innovative solutions to support humans life and work. To move towards this direction, in this dissertation, the Augmented Worlds vision has been presented and discussed. To author's best knowledge this work represents the first attempt proposed in literature to bring all together with the important purpose to provide not only a conceptual model for modelling the new generation of smart environments but also a concrete software platform to start playing and developing, abstracting from most of the technological issues and concentrating to the design of the application logic.

Although this last affirmation could appear trivial, considering the number of concepts involved in the AW vision and the heterogeneity of technologies required to design and develop such kind of novel environments, the opportunity to have a reference model and a platform based on it to approach is not to be underestimated.

Referring to related works and literature, the work presented in this dissertation provides several improvements to the state of the art of Pervasive Computing. Here, IoT concretely meets Mixed Reality with the adjunct goal to consider humans as a part of the technological loop. Previous works in that direction generally consider humans only as external users of software systems avoiding to give them an active role. Also, from the Mixed Reality side, most of the available approaches focus on the user experience as a solitary experience. Generally, the multi-user dimension and issues about cooperation and collaboration are not considered in available Mixed Reality software solutions, leaving the developer the responsibility to build an ad-hoc solution to solve this problem.

Although the research in this field and the engineering of the IoT have the great worth to let people today of living within an even more connected physical world bringing the computation outside computers and mobile devices, Vice versa, from the Pervasive Computing side we have never assisted to a deeper integration of digital reality with the physical one as in the approach proposed by augmented worlds, in particular, both on conceptual and practical dimensions. Moreover, agents and MAS has been introduced to propose a model where digital entities can be more proactive, reasoning and acting on the physical world, beyond the only digital/computational one. And this is another strong point of the approach.

To summing up, improvements to state of the art produced by studies and research by this dissertation are:

- an innovative solution to merge augmentation technologies both for humans and the environments in a unique frame of reference opening novel opportunities for building future hybrid smart environments;

- the formalization of an abstract conceptual meta-model defining the structure and the interaction dimensions of all elements and entities involved in the design of novel smart environments where augmentation is the main guideline;

- the availability of a common glossary to naming involved elements for future exploration in these directions to be used as a reference;

- the design and the development of a concrete prototype of a software infrastructure – the MiRAgE platform – to design an develop real instances of augmented worlds with the double opportunity to both evaluate the formal model and to start to develop hybrid smart environments abstracting from most of all involved technologies.

In particular, for the research in the Mixed Reality field, improvements are:

- the availability of a model enabling for the design of shared mixed reality environments, where collaboration and cooperation among human users is managed as a first-class feature;
- the availability of a platform for developing heterogeneous applications enabling Mixed Reality combined with several degrees of augmentation;
- a generic tool to integrate into the same digital environment several Mixed Reality devices and visors, escaping from technological issues that this objective entails;
- the opportunity to manage in the Mixed Reality also the physical environment and physical things;
- the opportunity to simply define proactive behaviours in terms of cognitive agents for holograms and digital inhabitants of a Mixed Reality oriented environment.

Vice versa, for the research on the Pervasive Computing and IoT side, this dissertation offers improvements in term of:

- the integration of the complete digital content and holograms as "thing" in the IoT and WoT vision, currently related only to real computational things;
- a platform for building pervasive shared and distributed software systems based on a common reference model integrating aspects coming from digital and real layers, beyond the integration of the Mixed Reality dimension.

Finally, also for Agents and MAS research, Augmented Worlds brings to state of the art significant advances as:

- the opportunity to build environments where to simulate agent societies exploiting features and data also coming from a real environment;
- opening the Artificial Intelligence area to applications strongly coupled with the real world, to apply and improve related techniques.

In spite of what was said in this section for an overall evaluation, this dissertation does not propose itself as an ending point for a work that integrates multiple concepts coming from heterogeneous research areas in the ICT field. Instead, we prefer to consider this work as a starting point for future explorations towards the realisation of the next generation of smart environments, as introduced in 4. This work is a kind of "starter-kit" both on the conceptual and practical side for reasoning about how concretely exploiting augmentation technologies to improve people lives, providing them new opportunities for working, cooperate and collaborate through new technologies and mediums.

Nevertheless, it also represents a point where a lot of convergences – for a long time considered only from their particular aspects and point of view – find the opportunity to be managed together in an integrated vision.

9.2 Critical Points and Limitations

Beyond from current MiRAgE prototype technological limitations briefly discussed in the previous chapter – that represent a first starting point for future works and explorations (see next session for further details) – in this section we want to discuss critical aspects of the complete vision presented in this dissertation.

Applicability. A principal limitation of the augmented world vision is about applicability. As reported in the first chapter of this dissertation, augmentation technologies are mature to be used out of the labs but, although we are not too far away to this, it is still too early to suppose that these technologies become part of everyday life. In spite of this, it is not to early to start to think of some particular workplaces as augmented worlds – for instance, within the application domains described in Chapter 6. We have to avoid to make the mistake of thinking that whichever environment could become an augmented world and identify considering also the real degree of impact on people dealing with that specific environment. Augmented worlds could be the keystone to develop next future smart environments, but it will be a gradual and slow process of innovation.

Augmenting avoiding overloading. Designing future smart environments as Augmented Worlds could leading into MR contexts where information and offered features to humans and agents, could have a negative impact to the whole systems in the case gives rises to overload rather than a real augmentation. Further studies in fields of Human-Computer Interactions and users interfaces in the context of mixed reality are required to obtain an adequate social acceptance of smart environments as in the vision of augmented worlds.

Availability of Mixed Reality technologies and devices. Mixed reality technologies are nowadays widely available but are still not so cheap and not so familiar to most people as required to think about a wide diffusion of augmented worlds. This is a critical point because to think about the design on new environments following this paradigm we have overpass people scepticism firstly. To deal with this, the requirement to support several degrees of augmented reality helps us: we can start to distribute augmentation functionalities exploiting a lower level of mixed reality trough smartphone that each person owns, moving to more sophisticated technologies in the following step. Another significant limitation is the battery life of most of the mixed reality enabling devices, which should be extended to allow users to make the most from the technology in daily use.

Ethics and Privacy. A serious concern about augmented worlds vision is related to the issue of privacy. In an augmented world the digital layer becomes like a twin for the physical one and having humans involved in the loop, agents can learn a lot of information and data about them. If we couple this opportunity with artificial intelligence and machine learning techniques we could build software systems able to violate ethics and people's privacy potentially. Augmented worlds should also involve data protection techniques, especially for agents that should become able to learn only the minimum amount of data from the environment required to deal with their goals. Finally, there's the argument of information overload which is often applied to technology in general. Many people argue that we live in a constantly switched on society and such a 24/7 technology on demand will radically change the way we see and think about reality. They often cite the danger of spending too much time immersed in the digital world and missing out on moments that are happening in the real world. This surely represents a critical point to be dealt with to spread the vision of augmented worlds as a way to augmented peoples and environment with a silver lining.

9.3 Future Works and Explorations

In the above overall evaluation, we have defined this work as a starting point of a new research direction in the smart environments area. As a starting point, many explorations require to be accomplished in the next future.

The Augmented Worlds approach represents an important reference model to design pervasive mixed reality software systems, but refinements to the model can be discussed and introduced after a more in-depth evaluation to it performed through the development of more complex and more challenging case studies to get proper feedbacks to that purpose. To develop such case studies, future work will be devoted to refine and improve the architecture and prototype implementation of MiRAgE, in particular, to make it widely available to be used as a possible reference platform for intelligent environments where the role of augmentation technologies is predominant.

Research directions that can be taken starting from the work discussed in this dissertation are many. Anyway, generally speaking, for what concerns the author of this dissertation, future works will be direct on a stronger development of MiRAgE to exit from the prototype stage and make it available for the community. We will consider other extensions in addition to the ones reported in Section 7.5. Furthermore, libraries to support other agents and MAS technologies will be developed. Finally, a stronger discussion about the approach will be tackled after the next steps of MiRAgE refactoring and further development.

Nevertheless, as a research product, MiRAgE and its related conceptual model are available to the community as open source technologies to encourage the community to undertake other direction starting from the current state of development of the infrastructure, even up to improve the conceptual model and meta-model of augmented worlds.

Conclusions

After the exploration of the scientific and research literature in the context of augmentation technologies – as they have been defined and described in Chapter 1 – emerged how these technologies have reached a high level of development. That is they are mature enough to be able to imagine scenarios where they could be used to reduce the existing gap perceived by humans between the digital and the physical world, between the reality in which humans live and the computational level enabled by software systems.

Starting from this claim, this dissertation has investigated how to put together two distinct scenarios, the first related to the area of the Pervasive Computing as a kind of environment augmentation and the second related to the area of the Mixed Reality as a kind of human augmentation. This integration has been addressed with the purpose to go towards the definition, the design and the engineering of the next future's smart environments where people will live and work along with more and more intelligent proactive software agents. In other words, smart environments where the physical and the digital matters are seamlessly coupled, where the software is able to augmented the physical reality, allowing humans to get involved into what in this dissertation we call Augmented Worlds.

The first part of this work has been dedicated to the definition of a conceptual meta-model for augmented worlds, where the central concept is encapsulated in the idea of populating an augmented world with many augmented entities representing full-fledged computational elements "living" in the digital world but coupled with a specific location of the real world and perceivable by humans through holograms.

In the augmented world meta-model, humans are considered in the loop as one of the first-class abstractions. As their "counterpart" in the digital worlds, proactive software cognitive agents has also been considered in the meta-model as autonomous entities that can act over augmented entities, observing their status and managing them. The proposed meta-model has been inspired by research in the context of environments for multi-agent systems and can be considered also an adding value even for agents and MASs literature offering new opportunities also for related research agenda.

In the vision of this dissertation, an augmented world is an instance of a pervasive mixed reality agent-based software systems shaping a smart environment where humans and agents can cooperate and collaborate exploiting augmented entities as a medium to observe and manage both physical and digital “things”.

According to the proposed meta-model, the second part of this work is related to design and develop a concrete infrastructure – called MiRAgE – for engineering augmented worlds. This infrastructure represents the main result of this dissertation, offering to the literature and the community the availability to exploit a tool to develop and execute new kinds of smart environment where pervasive computing and mixed reality are unified in a homogeneous scenario.

Opportunities obtained by the design and the development of MiRAgE are multiple. In particular, the chance to start developing concrete augmented worlds can be useful to trigger further explorations beyond existing work in the context of Smart Environments, Cyber-Physical Systems, Mixed and Augmented Reality, Artificial Intelligence and Multi-Agents Systems. Moreover, the broader notion of augmentation brought by augmented worlds through MiRAgE allows for developing a new generation of agent-based systems built upon a bidirectional augmentation of the physical and digital matter, in the sense that for software agents shaping and living in AWs, the physical environments can be used as an augmentation of their (virtual) environment. As humans perceive and act upon virtual augmented entities in their physical reality, agents living in the augmented layer (and controlling the augmented entities) can perceive and act upon physical entities, through a properly designed agent environment coupled to the physical worlds using pervasive computing technologies.

Considered application domains have emphasised how the AWs approach could be disruptive in several significant scenarios (among others, Industry 4.0 and Healthcare contexts). Furthermore, first developed case studies have demonstrated how the MiRAgE infrastructure could be very effective in the design of such smart environments, allowing a developer to abstracts from basic elements required from distributed, real-time and shared software systems – getting for free, among others, both the multi-users cooperative dimension and the holograms’ management – and mainly focus to the application logic of the systems.

Although the one proposed in this dissertation represent only a first but concrete step toward the realization of the augmented world vision, this work defines the basis for starting to concretely prototype a new generation of smart environments in which computation and programming can be exploited to shape various forms of augmentation of the physical world and beyond.

Of course, the research agenda in such direction is rich in terms of future works to be discussed and addressed. Firstly, starting from the improvement and the consolidation of the work done in this dissertation, oriented but not limited to make MiRAgE widely available to be used as a possible reference platform for developing future smart environments where the role of augmentation technologies is predominant. Secondly, promoting the development of case studies applying the AWs meta-model to obtain a stronger evaluation of the model itself and feedbacks to improve it. Finally, another important and ambitious objective is about expanding the community of people and researchers working in the contexts of mixed reality, pervasive computing and cognitive agents for continuing to discuss, share ideas and cooperate toward the vision in which augmented worlds could represent the solution for concretely reshaping people's ways of living, working and interacting in the next future.

References

[1] Abowd, G. D., Dey, A. K., Brown, P. J., Davies, N., Smith, M., and Steggles, P. (1999). Towards a better understanding of context and context-awareness. In *Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing*, HUC '99, pages 304–307, London, UK, UK. Springer-Verlag.

[2] Ashton, K. (2009). That "internet of things" thing: In the real world things matter more than ideas. *RID Journal*.

[3] Azuma, R. T. (1997). A survey of augmented reality. *Presence: Teleoper. Virtual Environ.*, 6(4):355–385.

[4] Barella, A., Carrascosa, C., Botti, V. J., and Martí, M. (2007). Multi-agent systems applied to virtual environments: a case study. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology, VRST 2007, Newport Beach, California, USA, November 5-7, 2007*, pages 237–238.

[5] Behrens, T. M., Hindriks, K. V., and Dix, J. (2011). Towards an environment interface standard for agent platforms. *Annals of Mathematics and Artificial Intelligence*, 61(4):261–295.

[6] Billinghurst, M. and Kato, H. (1999). Collaborative mixed reality. *Proceedings of the First International Symposium on Mixed Reality (ISMR'99). Mixed Reality – Merging Real and Virtual Worlds*, pages 261–284.

[7] Billinghurst, M. and Kato, H. (2002). Collaborative augmented reality. *Commun. ACM*, 45(7):64–70.

[8] Boissier, O., Bordini, R. H., Hübner, J. F., Ricci, A., and Santi, A. (2013). Multi-agent oriented programming with jacamo. *Science of Computer Programming*, 78(6):747–761.

[9] Bordini, R. H., Hübner, J. F., and Wooldridge, M. (2007). *Programming Multi-Agent Systems in AgentSpeak Using Jason (Wiley Series in Agent Technology)*. John Wiley & Sons, Inc., USA.

[10] Bratman, M. (1987). *Intention, plans, and practical reason.* Harvard University Press, Cambridge, MA.

[11] Brown, P. J., Bovey, J. D., and Chen, X. (1997). Context-aware applications: from the laboratory to the marketplace. *IEEE Personal Communications*, 4(5):58–64.

[12] Büscher, M., Christensen, M., Grønbæk, K., Krogh, P., Mogensen, P., Shapiro, D., and Orbæk, P. (2000). Collaborative augmented reality environments: Integrating vr, working materials, and distributed work spaces. In *Proceedings of the Third International Conference on Collaborative Virtual Environments*, CVE '00, pages 47–56, New York, NY, USA. ACM.

[13] Campbell, A. G., Stafford, J. W., Holz, T., and O'hare, G. M. (2014). Why, when and how to use augmented reality agents (auras). *Virtual Real.*, 18(2):139–159.

[14] Collier, R. W., Russell, S., and Lillis, D. (2015). Reflecting on agent programming with agentspeak(l). In Chen, Q., Torroni, P., Villata, S., Hsu, J., and Omicini, A., editors, *PRIMA 2015: Principles and Practice of Multi-Agent Systems*, pages 351–366, Cham. Springer International Publishing.

[15] Cook, D. and Das, S. (2004). *Smart Environments: Technology, Protocols and Applications (Wiley Series on Parallel and Distributed Computing).* Wiley-Interscience, New York, NY, USA.

[16] Craig, A. B. (2013). *Understanding Augmented Reality: Concepts and Applications.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edition.

[17] Croatti, A. and Ricci, A. (2017a). Mashing up the physical and augmented reality: The web of augmented things idea. In *Proceedings of the Eighth International Workshop on the Web of Things*, WoT 2017, pages 4–7, New York, NY, USA. ACM.

[18] Croatti, A. and Ricci, A. (2017b). Towards the web of augmented things. In *2017 IEEE International Conference on Software Architecture Workshops (ICSAW)*, pages 80–87. IEEE.

[19] Croatti, A. and Ricci, A. (2018a). Developing agent-based pervasive mixed reality systems: The mirage framework. In Demazeau, Y., An, B., Bajo, J., and Fernández-Caballero, A., editors, *Advances in Practical Applications of Agents, Multi-Agent Systems, and Complexity: The PAAMS Collection*, pages 301–304, Cham. Springer International Publishing.

[20] Croatti, A. and Ricci, A. (2018b). A model and platform for building agent-based pervasive mixed reality systems. In Demazeau, Y., An, B., Bajo, J., and Fernández-Caballero, A., editors, *Advances in Practical Applications of Agents, Multi-Agent Sys-*

tems, and Complexity: The PAAMS Collection, pages 127–139, Cham. Springer International Publishing.

[21] Daily, M., Oulasvirta, A., and Rekimoto, J. (2017). Technology for human augmentation. *Computer*, 50(2):12–15.

[22] Dautenhahn, K. (1999). Embodiment and interaction in socially intelligent life-like agents. In Nehaniv, C. L., editor, *Computation for Metaphors, Analogy, and Agents*, pages 102–141. Springer Berlin Heidelberg, Berlin, Heidelberg.

[23] Dey, A. K. (1998). Context-aware computing: The cyberdesk project. Technical report, AAAI 1998 Spring Symposium on Intelligent Environments.

[24] Eckstein, B. and Lugrin, B. (2016). Augmented reasoning in the mirror world. In *Proceedings of the 22Nd ACM Conference on Virtual Reality Software and Technology*, VRST '16, pages 313–314, New York, NY, USA. ACM.

[25] Engelbart, D. C. (1962). Augmenting human intellect : A conceptual framework. `http://www.1962paper.org/web.html`. Accessed: 2018-08-10.

[26] Engelbart, D. C. and English, W. K. (1968). A research center for augmenting human intellect. In *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I*, AFIPS '68 (Fall, part I), pages 395–410, New York, NY, USA. ACM.

[27] Ferber, J. (1999). *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition.

[28] Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine. AAI9980887.

[29] Gelernter, D. (1985). Generative communication in linda. *ACM Trans. Program. Lang. Syst.*, 7(1):80–112.

[30] Gelernter, D. (1991). *Mirror Worlds or the Day Software Puts the Universe in a Shoebox: How Will It Happen and What It Will Mean*. Oxford University Press, Inc., New York, NY, USA.

[31] Georgeff, M. P., Pell, B., Pollack, M. E., Tambe, M., and Wooldridge, M. (1999). The belief-desire-intention model of agency. In *Proceedings of the 5th International Workshop on Intelligent Agents V, Agent Theories, Architectures, and Languages*, ATAL '98, pages 1–10, London, UK. Springer-Verlag.

[32] Grubert, J., Langlotz, T., Zollmann, S., and Regenbrecht, H. (2017). Towards pervasive augmented reality: Context-awareness in augmented reality. *IEEE Transactions on Visualization and Computer Graphics*, 23(6):1706–1724.

[33] Guinard, D. and Trifa, V. (2016). *Building the Web of Things: With Examples in Node.Js and Raspberry Pi.* Manning Publications Co., Greenwich, CT, USA, 1st edition.

[34] Holz, T., Campbell, A. G., O'Hare, G. M. P., Stafford, J. W., Martin, A., and Dragone, M. (2011). Mira-mixed reality agents. *Int. J. Hum.-Comput. Stud.*, 69(4):251–268.

[35] Kimber, D., Vaughan, J., and Rieffel, E. (2011). Augmented perception through mirror worlds. In *Proceedings of the 2Nd Augmented Human International Conference*, AH '11, pages 36:1–36:2, New York, NY, USA. ACM.

[36] Kurkovsky, S. (2007). Pervasive computing: Past, present and future. In *2007 ITI 5th International Conference on Information and Communications Technology*, pages 65–71.

[37] Luck, M. and Aylett, R. (2000). Applying artificial intelligence to virtual reality: Intelligent virtual environments. *Applied Artificial Intelligence*, 14(1):3–32.

[38] Madakam, S., Ramaswamy, R., and Tripathi, S. (2015). Internet of things (iot): A literature review. *Journal of Computer and Communications*, 3:164–173.

[39] Milgram, P. and Kishino, F. (1994). A taxonomy of mixed reality visual displays. *IEICE Trans. Information Systems*, E77-D(12):1321–1329.

[40] Moreno-Cano, M. V., Santa, J., Zamora-Izquierdo, M. A., and Skarmeta, A. F. (2015). Future human-centric smart environments. *Modeling and Processing for Next-Generation Big-Data Technologies: With Applications and Case Studies*, pages 341–365.

[41] Oh, S., Woo, W., vr Lab, G. U., and Korea, S. (2009). Camar: Context-aware mobile augmented reality in smart space.

[42] Omicini, A., Ricci, A., and Viroli, M. (2008). Artifacts in the a&a meta-model for multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 17(3):432–456.

[43] Parrott, A. and Lane, W. (2017). Industry 4.0 and the digital twin: Manufacturing meets its match. Deloitte University Press.

[44] Pell, M. (2017). *Envisioning Holograms. Design Breakthrough Experiences for Mixed Reality.* Apress.

[45] Piekarski, W. and Thomas, B. (2002). Arquake: The outdoor augmented reality gaming system. *Commun. ACM*, 45(1):36–38.

[46] Poovendran, R. (2010). Cyber–physical systems: Close encounters between two parallel worlds [point of view]. *Proceedings of the IEEE*, 98(8):1363–1366.

[47] Rao, A. S. and Georgeff, M. P. (1995). Bdi agents: From theory to practice. In *Proceedings of the 1st international conference on multi-agent systems (ICMAS-95*, pages 312–319.

[48] Reeves, S. (2012). Envisioning ubiquitous computing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 1573–1582. ACM.

[49] Ricci, A., Piunti, M., Tummolini, L., and Castelfranchi, C. (2015a). The mirror world: Preparing for mixed-reality living. *IEEE Pervasive Computing*, 14(2):60–63.

[50] Ricci, A., Tummolini, L., Piunti, M., Boissier, O., and Castelfranchi, C. (2015b). Mirror worlds as agent societies situated in mixed reality environments. *Coordination, Organizations, Institutions, and Norms in Agent Systems X: COIN 2014 International Workshops, COIN@AAMAS, Paris, France, May 6, 2014, COIN@PRICAI, Gold Coast, QLD, Australia, December 4, 2014, Revised Selected Papers*, pages 197–212.

[51] Ricci, A., Viroli, M., and Omicini, A. (2007). CArtAgO: A framework for prototyping artifact-based environments in MAS. In Weyns, D., Parunak, H. V. D., and Michel, F., editors, *Environments for MultiAgent Systems III*, volume 4389 of *LNAI*, pages 67–86. Springer.

[52] Ricci, A., Viroli, M., and Omicini, A. (2008). The a&a programming model and technology for developing agent environments in mas. In Dastani, M., El Fallah Seghrouchni, A., Ricci, A., and Winikoff, M., editors, *Programming Multi-Agent Systems*, pages 89–106, Berlin, Heidelberg. Springer Berlin Heidelberg.

[53] Rincon, J. A., Garcia, E., Julian, V., and Carrascosa, C. (2014). Developing adaptive agents situated in intelligent virtual environments. In Polycarpou, M., de Carvalho, A. C. P. L. F., Pan, J.-S., Woźniak, M., Quintian, H., and Corchado, E., editors, *Hybrid Artificial Intelligence Systems*, pages 98–109, Cham. Springer International Publishing.

[54] Rincon, J. A., Poza-Lujan, J. L., Julian, V., Posadas-Yague, J. L., and Carrascosa, C. (2016). Extending mam5 meta-model and JaCalIVE framework to integrate smart devices from real environments. *PLOS ONE*, 11(2):1–27.

[55] Rousseau, J. (2016). The laws of mixed reality, or mixed reality without the rose-colored glasses. `https://www.artefactgroup.com/articles/mixed-reality-without-rose-colored-glasses/`. Accessed: 2018-08-10.

[56] Satyanarayanan, M. (2001). Pervasive computing: vision and challenges. *IEEE Personal Communications*, 8(4):10–17.

[57] Schilit, B. N. and Theimer, M. M. (1994). Disseminating active map information to mobile hosts. *IEEE Network*, 8(5):22–32.

[58] Schmalstieg, D. and Höllerer, T. (2015). *Augmented Reality: Principles and Practice*. Addison-Wesley.

[59] Shekhar, S., Feiner, S. K., and Aref, W. G. (2015). Spatial computing. *Commun. ACM*, 59(1):72–81.

[60] Shin, C., Lee, W., Suh, Y., Yoon, H., Lee, Y., and Woo, W. (2009). Camar 2.0: Future direction of context-aware mobile augmented reality. In *2009 International Symposium on Ubiquitous Virtual Reality*, pages 21–24.

[61] Shoham, Y. (1993). Agent-oriented programming. *Artif. Intell.*, 60(1):51–92.

[62] Smith, D. A., Kay, A., Raab, A., and Reed, D. P. (2003). Croquet - a collaboration system architecture. In *First Conference on Creating, Connecting and Collaborating Through Computing, 2003. C5 2003. Proceedings.*, pages 2–9.

[63] Starner, T. (2013). Project glass: An extension of the self. *IEEE Pervasive Computing*, 12(2):14–16.

[64] Weiser, M. (1999). The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun. Rev.*, 3(3):3–11.

[65] Weiser, M., Gold, R., and Brown, J. S. (1999). The origins of ubiquitous computing research at parc in the late 1980s. *IBM Systems Journal*, 38(4):693–696.

[66] Weiss, G., editor (1999). *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. MIT Press, Cambridge, MA, USA.

[67] Wellner, P., Mackay, W., and Gold, R. (1993). Back to the real world. *Commun. ACM*, 36(7):24–26.

[68] Weyns, D., Omicini, A., and Odell, J. (2007). Environment as a first class abstraction in multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 14(1):5–30.

[69] Wooldridge, M. (2009). *An Introduction to MultiAgent Systems*. Wiley Publishing, 2nd edition.

[70] Wooldridge, M. and Jennings, N. R. (1995). Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 10:115–152.

[71] Xia, C. and Maes, P. (2013). The design of artifacts for augmenting intellect. In *Proceedings of the 4th Augmented Human International Conference*, AH '13, pages 154–161, New York, NY, USA. ACM.

[72] Zambonelli, F. and Mamei, M. (2005). Spatial computing: An emerging paradigm for autonomic computing and communication. In Smirnov, M., editor, *Autonomic Communication*, pages 44–57, Berlin, Heidelberg. Springer Berlin Heidelberg.