Interaction with Three Dimensional Virtual Humans in Ambient Intelligence Environments

Birliraki Chryssi

Thesis submitted in partial fulfillment of the requirements for the

Masters' of Science degree in Computer Science

University of Crete
School of Sciences and Engineering
Computer Science Department
Voutes Campus, Heraklion, GR-70013, Greece

Thesis Advisor: Prof. *Constantine Stephanidis*

This work has been supported by the Institute of Computer Science of the Foundation for Research and Technology-Hellas (FORTH).

UNIVERSITY OF CRETE

COMPUTER SCIENCE DEPARTMENT

Interaction with Three Dimensional Virtual Humans in Ambient Intelligence Environments

Thesis submitted by

BIRLIRAKI CHRYSSI

in partial fulfillment of the requirements for the
Masters' of Science degree in Computer Science

THESIS APPROVAL

Author: _____
Birliraki Chryssi

Committee Approvals:

Supervisor _____
Constantine Stephanidis
Professor

Member _____
George Papagiannakis
Assistant Professor

Member _____
Dimitris Grammenos
Principal Researcher

Department Approval: _____
Angelos Bilas
Professor, Director of Graduate Studies

Heraklion, March 2014

To my family.

Abstract

Virtual humans are embodied agents that look like, act like, and interact with humans but exist in virtual environments. They have the ability to communicate with the environment, as well as with other agents and users using both verbal and non-verbal means. Virtual Humans can be used to provide assistance, guidance, information and training in various application domains including education, cultural heritage, healthcare, tourism and everyday life activities. They may act either complimentary or as the primary means for information provision and can be instantiated in any available platform of the smart environment, also including personal computers and mobile devices. In the context of Ambient Intelligence and Human-Computer Interaction, they are a metaphor that makes systems more anthropocentric, in the sense that users have the impression that they interact with other humans rather than a computer system.

This thesis reports on the design, the development and the evaluation of a framework which provides a dynamic data modeling mechanism for storage and retrieval, implements virtual humans behaviors by creating body gestures and speech synthesis that can be used for information provision, creates interactive multimedia information visualizations (e.g. images, text, audio, videos, 3D models) and implements communication through multimodal interaction techniques. The interaction may involve human to agent, agent to environment or agent to agent communication. The framework supports alternative roles for the virtual agents who may act as assistants for existing systems, standalone “applications” or even as integral parts of emerging smart environments.

As assisting mechanisms (i.e., help systems), agents are able to aid users in learning to interact with a system, as well as to offer structured information which may help the users acquire an overall understanding of how a system works and what can be expected of it. This information can be presented through images, videos, text, audio, and three dimensional objects, as well as synthesized speech and body movements. Additionally, systems can take advantage of the agent in order to provide their users with real time interactive assistance. As a standalone “application”, structured information and interactive help about the interaction techniques are presented through the virtual humans. As a component of a smart environment, the virtual humans are able to perform both of the above functionalities in combination and separately, while also communicating with the environment and its participants (users and other systems).

The developed framework supports multiple multimodal interaction techniques, even in combination, so as to offer natural interaction in a wide range of hardware set-ups employing various output and input devices. Apart from the traditional desktop interaction techniques such as mouse and touch screens, additional approaches, specifically directed to navigation and manipulation in 3D virtual environments, were studied and implemented. Users can communicate with the system through gestures, using, that is, their bare hands to interact with items. Furthermore, users can use their voice for verbal interaction with the system and finally, users can apply gestures through mobile devices in order to manipulate the visualized information.

An evaluation study was conducted with the participation of 10 people to study the developed system in terms of usability and effectiveness, when it is employed as an assisting mechanism for another application. The evaluation results were highly positive and promising, confirming the system's usability and encouraging further research in this area.

Περίληψη

Οι ανθρωπόμορφοι εικονικοί χαρακτήρες είναι οντότητες που μοιάζουν, συμπεριφέρονται και αλληλεπιδρούν σαν άνθρωποι αλλά υπάρχουν σε εικονικά περιβάλλοντα. Έχουν τη δυνατότητα να επικοινωνήσουν με το περιβάλλον, με άλλους χαρακτήρες και με χρήστες χρησιμοποιώντας τόσο λεκτική όσο και μη λεκτική έκφραση. Οι ανθρωπόμορφοι εικονικοί χαρακτήρες μπορούν να χρησιμοποιηθούν για την παροχή βοήθειας, καθοδήγησης, πληροφορίας και πρακτικής εκπαίδευσης σε διάφορους τομείς εφαρμογής όπως η εκπαίδευση, ο πολιτισμός, η υγεία, ο τουρισμός και οι καθημερινές δραστηριότητες. Μπορεί να δρουν είτε συμπληρωματικά είτε ως ο κύριος τρόπος παροχής βοήθειας και μπορούν να ενσωματωθούν σε οποιαδήποτε διαθέσιμη πλατφόρμα ενός έξυπνου περιβάλλοντος, συμπεριλαμβανομένων των προσωπικών υπολογιστών και των φορητών συσκευών. Στο πλαίσιο της Διάχυτης Νοημοσύνης και της Επικοινωνίας Ανθρώπου-Μηχανής, καθιστούν τα συστήματα πιο ανθρωποκεντρικά με την έννοια ότι οι χρήστες έχουν την αίσθηση ότι δεν αλληλεπιδρούν με ένα σύστημα αλλά με έναν άλλο άνθρωπο.

Σε αυτό το πλαίσιο, η παρούσα εργασία αναλύει το σχεδιασμό, την υλοποίηση και την αξιολόγηση ενός πλαισίου εφαρμογών (framework) το οποίο παρέχει έναν μηχανισμό για την δυναμική αποθήκευση και ανάκτηση δεδομένων, υλοποιεί συμπεριφορές μέσω της δημιουργίας χειρονομιών και της χρήσης συνθετικής ομιλίας που μπορούν να χρησιμοποιηθούν για την παροχή πληροφορίας, δημιουργεί διαδραστικές απεικονίσεις πολυμεσικής πληροφορίας και επικοινωνεί μέσω πολυτροπικών μορφών αλληλεπίδρασης. Η αλληλεπίδραση περιλαμβάνει επικοινωνία ανάμεσα σε χρήστη και εικονική οντότητα, περιβάλλον και εικονική οντότητα καθώς και ανάμεσα σε εικονικές οντότητες. Το πλαίσιο εφαρμογών υποστηρίζει διαφορετικούς ρόλους για τους εικονικούς χαρακτήρες, οι οποίοι μπορούν να έχουν το ρόλο του βοηθού σε υπάρχοντα συστήματα, ξεχωριστές εφαρμογές ή ως βασικό κομμάτι σε έξυπνα περιβάλλοντα.

Ως βοηθητικά συστήματα, οι εικονικοί χαρακτήρες έχουν τη δυνατότητα να βοηθήσουν τη διαδικασία εκμάθησης της αλληλεπίδρασης με ένα σύστημα καθώς και να προσφέρουν δομημένη πληροφορία που μπορεί να βοηθήσει το χρήστη να αποκτήσει μία γενική εικόνα της λειτουργίας ενός συστήματος και των δυνατοτήτων του. Η πληροφορία που παρέχεται μπορεί να είναι εικόνες, κείμενο, ήχος, βίντεο και τρισδιάστατα μοντέλα, όπως και συνθετική ομιλία και κινήσεις του σώματος. Επιπλέον, τα συστήματα μπορούν να παρέχουν αλληλεπιδραστική βοήθεια σε πραγματικό χρόνο. Ως ξεχωριστές εφαρμογές, οι

ανθρωπόμορφοι εικονικοί χαρακτήρες μπορούν να παρέχουν δομημένη πληροφορία και αλληλεπιδραστική βοήθεια σχετικά με τις τεχνικές αλληλεπίδρασης τις οποίες μπορούν να χρησιμοποιήσουν οι χρήστες για να χειριστούν αυτές τις εφαρμογές. Ως μέρος ενός έξυπνου περιβάλλοντος μπορούν να έχουν και τις δύο προαναφερθείσες λειτουργικότητες, τόσο ανεξάρτητα όσο και συνδυαστικά, ενώ επίσης μπορούν να επικοινωνούν με το περιβάλλον (χρήστες και άλλα συστήματα).

Το πλαίσιο εφαρμογών που αναπτύχθηκε είναι σχεδιασμένο ώστε να υποστηρίζει πολλαπλές πολυτροπικές τεχνικές αλληλεπίδρασης, οι οποίες μπορούν να χρησιμοποιηθούν ακόμα και συνδυαστικά, έτσι ώστε να προσφέρουν φυσική αλληλεπίδραση σε ένα ευρύ φάσμα εγκαταστάσεων. Εκτός από τους τρόπους αλληλεπίδρασης που βασίζονται σε επιτραπέζια συστήματα (όπως το ποντίκι και η οθόνη αφής), εναλλακτικές προσεγγίσεις μελετήθηκαν και υλοποιήθηκαν στοχεύοντας συγκεκριμένα στην πλοήγηση και το χειρισμό τρισδιάστατων περιβαλλόντων. Οι χρήστες μπορούν να επικοινωνήσουν με το σύστημα μέσω χειρονομιών χρησιμοποιώντας δηλαδή τα χέρια τους αλλά και με τη φωνή τους ώστε να μπορούν να αλληλεπιδράσουν λεκτικά με το σύστημα. Τέλος, οι χρήστες μπορούν να χρησιμοποιήσουν χειρονομίες μέσω φορητών συσκευών, προκειμένου να χειριστούν την πληροφορία που εμφανίζεται.

Μια μελέτη αξιολόγησης πραγματοποιήθηκε με τη συμμετοχή 10 ατόμων για να αξιολογήσει το σύστημα που αναπτύχθηκε ως προς τη ευχρηστία και την αποτελεσματικότητα όταν χρησιμοποιείται ως μηχανισμός βοήθειας για μια άλλη εφαρμογή. Τα αποτελέσματα της αξιολόγησης ήταν ιδιαίτερα θετικά και ελπιδοφόρα, επιβεβαιώνοντας τη χρηστικότητα του συστήματος και την ενθάρρυνση για περαιτέρω έρευνα σε αυτόν τον τομέα.

Acknowledgements

I would like to thank my supervisor Prof. Constantine Stephanidis for the support and encouragement throughout the conduction of this thesis.

Furthermore, I would like to thank my colleagues at the Human-Computer Interaction laboratory of ICS-FORTH, especially Dr. Dimitris Grammenos, Ilia Adami and Anthony Katzourakis for the graphics work.

The work reported in this thesis has been conducted in the context of a scholarship provided by the Institute of Computer Science (ICS) of the Foundation for Research and Technology – Hellas (FORTH), including financial support and the use of the necessary equipment facilities.

I would like to thank all my friends who where by my side during this thesis, but especially I would like to thank Giannis who helped me in every phase of this work for his caring, support and patience.

Additionally, I would like to thank my parents, Manolis and Maria, as well as Aspa, George, my grandmother Chrysoula, Stelios and Vicky, who despite the distance between us stood by me at all times. Last but not least, I would like to thank Zozetta for her invaluable help.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επόπτη μου Καθηγητή κ. Κωνσταντίνο Στεφανίδη για την υποστήριξη και την ενθάρρυνση καθ 'όλη τη διεξαγωγή της παρούσας διατριβής.

Επιπλέον , θα ήθελα να ευχαριστήσω τους συναδέλφους μου στο εργαστήριο Αλληλεπίδρασης Ανθρώπου - Υπολογιστών του ΙΠ- ΙΤΕ , ιδιαίτερα το Δόκτωρ Δημήτρη Γραμμένο, την Ηλία Αδάμη και τον Αντώνη Κατζουράκη για τη γραφιστική εργασία.

Οι εργασία που αναφέρεται στην παρούσα διατριβή έχει διεξαχθεί στο πλαίσιο μιας υποτροφίας που παρέχεται από το Ινστιτούτο Πληροφορικής (ΙΠ) του Ιδρύματος Τεχνολογίας και Έρευνας - Ελλάς (ΙΤΕ) , συμπεριλαμβανομένης της οικονομικής ενίσχυσης και τη χρήση του αναγκαίου εξοπλισμού.

Θα ήθελα να ευχαριστήσω όλους τους φίλους μου οι οποίοι βρίσκονταν στο πλευρό μου κατά τη διάρκεια αυτής της εργασίας , αλλά ιδιαίτερα θα ήθελα να ευχαριστήσω τον Γιάννη που με βοήθησε σε κάθε φάση αυτού του έργου και κυρίως για τη φροντίδα , την υποστήριξη και την υπομονή του.

Επίσης , θα ήθελα να ευχαριστήσω τους γονείς μου , Μανώλη και Μαρία , καθώς και τα αδέρφια μου Άσπα και Γιώργο , τη γιαγιά μου Χρυσούλα , το Στέλιο και τη Βίκυ , οι οποίοι παρά την απόσταση μεταξύ μας στάθηκαν δίπλα μου όλες τις στιγμές. Τελευταία, αλλά όχι λιγότερο σημαντική , θα ήθελα να ευχαριστήσω τη Ζωζέτα για την πολύτιμη βοήθειά της.

Table of Contents

Contents

List of Figures

List of Diagrams

1 Introduction

Virtual humans are embodied agents that exist in virtual environments that look and act like humans and can interact with them. They are employed as user interfaces and can serve various needs for human-computer interaction, including guidance, assistance, information provision and user training. Virtual humans exhibit human-like qualities and can communicate with humans or even with each other using natural human modalities and are capable of real-time perception, cognition and action.

Ambient Intelligence (AmI) environments are characterized by the ubiquitous and unobtrusive presence of electronic equipment in the users' environment that allows sensing the users' actions and react accordingly in order to help them achieve their goals. In AmI environments the interface is indistinguishable from the physical setting, as the real world assumes the role of the interface.

The incorporation of virtual humans in ambient intelligence environments can enhance the social aspects of interaction offering human-like anthropocentric communication.

This thesis presents the design, the development and the evaluation of a framework which employs virtual humans able to provide multimodal interaction, assistance and information provision in AmI environments. In general, the framework:

- Uses a Data Model for information storage and retrieval
- Is dynamic and flexible so as to fit diverse needs of other systems and the environment
- Allows virtual humans to act as assistants to other systems offering real-time help, tutorials and user training on interaction techniques
- Offers the ability to use virtual humans as standalone "applications"
- Uses the virtual human's body gestures and speech as a way of information provision
- Is able to visualize information in different forms, such as images, videos, 3D models, text and audio
- Supports natural multimodal interaction using a variety of means including verbal interaction and gestures
- Integrates techniques regarding user interaction in three dimensions and mobile devices

This thesis is structured as follows:

Chapter 2 presents related work regarding the different aspects of the thesis. Existing approaches that are related to the characteristics of Virtual Human systems are presented, analyzed and discussed. Each proposed approach is analyzed highlighting its related advantages and disadvantages.

Chapter 3 describes the research objectives of this thesis depicting the motivation and rationale behind of the conceptualization, design and implementation of the suggested framework, along with its innovative aspects.

Chapter 4 examines the requirements regarding the presented work. The process of requirements elicitation is described step-by-step, followed by the results concerning the user requirements of the final outcome.

Chapter 5 provides an overview of the framework as an assisting tool. Firstly, fundamental concepts are investigated and analyzed in order to allow the reader to perceive the basic terms of the framework. Moreover, the Data Modeling for storage and retrieval is presented in respect to the aforementioned terms. Furthermore, the design of the system is presented and all the decisions that were taken in this process. Additionally, a Hierarchical Task Analysis is shown related to user actions. Finally, scenarios of use are described in order to ascribe the necessity of virtual humans in ambient intelligence environments and to showcase studies where virtual humans are used, as assistants, as standalone applications or as parts of other smart systems in different roles. The advantages and disadvantages of their presence in such environments are described too.

Chapter 6 examines the natural multimodal interaction methods that are supported. Each interaction technique is thoroughly analyzed according to the available actions that the user can take in order to manipulate the system.

Chapter 0 describes the implementation details of the framework outlining its individual functional modules and their roles.

Chapter 8 discusses the evaluation process and its results regarding the usability of the system and its effectiveness as an assistance provision mechanism for third-party systems.

Chapter 9 presents the conclusions in respect to the initial objectives, taking into consideration the results of the evaluation process. Furthermore, future work is discussed

concerning additions that can improve the developed system and enhance the overall user experience.

2 Background and Related Work

2.1 Ambient Intelligence

2.1.1 Ambient Intelligence Environments

Ambient intelligence is an emerging discipline that brings intelligence to our everyday environments and makes those environments sensitive to us. Ambient intelligence (AmI) research builds upon advances in sensors and sensor networks, pervasive computing, and artificial intelligence. Since these contributing fields have experienced tremendous growth in the last few years, AmI research has strengthened and expanded: ambient intelligence environments are maturing and the resulting technologies promise to revolutionize daily human life by making people's surroundings flexible and adaptive.

As mentioned in [7], several definitions for Ambient Intelligence are presented in literature, including:

1. “A developing technology that will increasingly make our everyday environment sensitive and responsive to our presence”
2. “A potential future in which we will be surrounded by intelligent objects and in which the environment will recognize the presence of persons and will respond to it in an undetectable manner”
3. “Ambient Intelligence implies intelligence that is all around us”
4. “The presence of a digital environment that is sensitive, adaptive, and responsive to the presence of people”
5. “ A vision of future daily life… contains the assumption that intelligent technology should disappear into our environment to bring humans an easy and entertaining life”
6. “A new research area for distributed, non-intrusive, and intelligent software systems”
7. “In an AmI environment people are surrounded with networks of embedded intelligent devices that can sense their state, anticipate, and perhaps adapt to their needs”

8. “A digital environment that supports people in their daily lives in a nonintrusive way”

All the above highlight the features that are expected in Ami environments and technologies: **sensitivity**, **adaptation**, **transparency**, **ubiquity** and **intelligence**. From these definitions and the features that are used to characterize Ami we can see how the discipline compares and contrasts with fields such as pervasive computing, ubiquitous computing, and artificial intelligence. The fact that AmI systems must be sensitive, responsive, and adaptive highlights the dependence that AmI research has on context-aware computing. Similarly, the AmI characteristic of transparency is certainly aligned with the concept of the disappearing computer. This methodological trend was envisioned by Weiser [1], who stated: ``The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it''.

As a result of the above features and definitions comes one general definition which states that an Ambient Intelligence system is a digital environment that proactively, but sensibly, supports people in their daily lives.

From its definition, we can see that Ambient Intelligence has a decisive relationship with many areas in computer science. The contributing technologies are organized into five areas, shown in the figure below (Fig. 1). A key factor in AmI research is the presence of intelligence. As such, the AmI algorithm perceives the state of the environment and users with sensors, reasons about the data using a variety of AI techniques, and acts upon the environment using controllers in such a way that the algorithm achieves its intended goal.

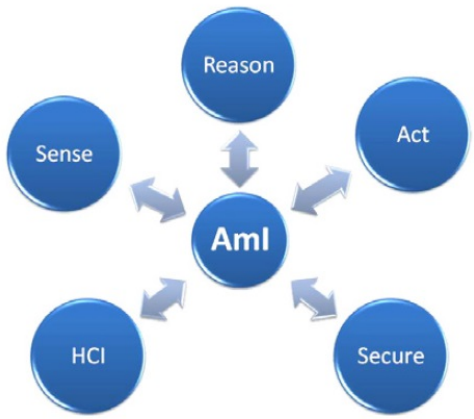


Fig. 1: Relationship between Ami and contributing technologies.

An important aspect of AmI has to do with interactivity [5]. On the one hand there is a motivation to reduce explicit human-computer interaction (HCI) as the system is supposed to use its intelligence to infer the situations and user needs from the observed activities, as if a passive human assistant were observing the activities unfold with the expectation to help when (and only if) required. Systems are expected to provide situation-aware information (I-want-here-now) through natural interfaces. On the other hand, a diversity of users may need or voluntarily seek direct interaction with the system to indicate preferences, needs, etc. HCI has been an important area of computer science since the inception of computing as an area of study and development. Today, with so many gadgets incorporating computing power of some sort, HCI continues to thrive as an important AmI topic. [2]

It is also worthwhile to note that the concept of AmI is closely related to the "service science" in the sense that the objective is to offer proper services to users. It may not be of interest to a user what kinds of sensors are embedded in the environment or what type of middleware architecture is deployed to connect them. Only the services given to the user matter to them. Therefore, the main thrust of research in AmI should be integration of existing technologies rather than development of each elemental device. [6]

Within the notion of the service science, it is important to consider the implications of user evaluation in a "service-evaluation-research" loop (Fig. 2). As part of their investigation, AmI researchers should assess efforts to setting up working intelligent environments in which users conduct their normal activities. As the system is distributed in the environment, users only notice services they receive. The researchers then observe the interaction and (re-)evaluate the system. The result of the evaluation may lead to identifying new services, new viewpoints or even new evaluation criteria. Then all of the findings can be fed back to new research and development.

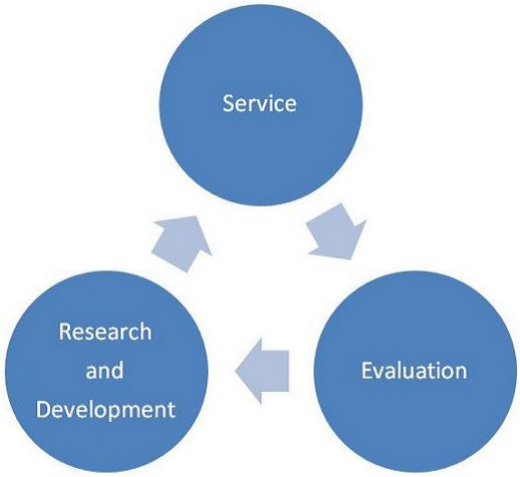


Fig. 2: Service-Evaluation-Research

2.1.2 Ambient Intelligence Interaction Techniques

In Ambient Intelligence environments the interaction is different from the typical interaction techniques (mouse, keyboard etc.), due to the fact that AmI environments should be "transparent" from the technological point of view and their interaction techniques are human-centric, so as humans are able to react naturally and feel comfortably during use. Multimodality is the "key" for AmI to provide systems that can be easily manipulated from different groups of people who will operate in many different ways individually. Several multimodal techniques appear in literature, each with their advantages and disadvantages, which were studied and analyzed for the implementation of this thesis. Ambient Intelligence allows the user to interact with several means often simultaneously, such as speech, body movements, gestures, eye and head tracking or mobile phones (smart phones).

In its most basic sense, **multimodality** is the mixture of textual, audio, and visual modes in combination with media and materiality to create meaning [3]. In recent years, multimodal interfaces have gained momentum as an alternative to traditional WIMP ("windows-icons-menus-pointers") interaction styles. People more naturally interact in a multimodal way with the world, through both parallel and sequential use of multiple perceptual modalities [8]. Human–Computer Interaction has sought for decades to provide computers with similar capabilities, in order to provide more natural, powerful, and persuasive interactive experiences. With the rapid advance in non-desktop computing,

generated by powerful mobile devices and affordable sensors in recent years, multimodal research that leverages speech, touch, vision, and gesture is on the rise.

It is worth mentioning that breakthroughs in hardware setups influence trends as new opportunities arise by innovative features that may be introduced. For instance, recognition of objects and humans used to be a difficult, processor demanding and expensive task. However, Microsoft's Kinect [9] included the employment of a laser sensor with a depth camera as a supplement of a normal RGB camera, providing a low cost – around 150 euro, accurate and stable input device. Yet, despite hardware evolution dependence, Ambient Intelligence aims to develop and provide all the necessary techniques for natural interaction in a variety of means as hardware evolution is considered a given.

- Voice Interaction through speech recognition

In computer science speech recognition is the translation of spoken words into text. Some systems use "speaker independent speech recognition", while others use "training" where an individual speaker reads sections of text into the system, like Hindi speech recognition system or Kaldi Speech Recognition System both using HTK [15,16,74]. Speech is one of the most natural forms of communication between humans, although between machines and humans for quite a few it is awkward, because a lot of them are not familiar to technological news and the fact that a man can possibly have a conversation with a machine scares them. On the other hand, the feature of invisibility in speech is vital because in ambient intelligence environments the transparency is important. This make the technique challenging to voice-based interfaces in the design and use, but simultaneously helps them to provide an alternative way of interaction with impaired users or in environments where other techniques are difficult to be used or cannot be used at all.

- Gesture-based interaction

Definitions:

- *A gesture is a motion of the body that contains information. Waving goodbye is a gesture. Pressing a key on a keyboard is not a gesture because the motion of a finger on its way to hitting a key is neither observed nor significant. All that matters is which key was pressed.[17]*

- *Any physical movement that a digital system can sense and respond to without the aid of a traditional pointing device such as a mouse or stylus. A wave, a head nod, a touch, a toe tap, and even a raised eyebrow can be a gesture [18]*

Gestures can be defined as a form of non-verbal communication in which visible body actions communicate particular messages. The recognition of gestures is an issue on which many work exists in literature [13, 12]. and it is an area that computer's science future such as many other sciences, like social science, will be interested in.

The current peak in gesture-based interaction started with game consoles, like Wii in 2006 or Microsoft Kinect and Asus Xtion in 2010. Nowadays, gesture identification is robust enough [14] to let developers create applications which promotes user's communication with the systems. There is a diversity of gesture types such as hand gestures [69] , body gestures[46]. Those gestures should always be used with discretion in every application because their goal is to help and not to confuse and fatigue the interaction of the users with the system. So, they can operate their actions supplementary to other techniques such as mouse events.

Gestures can also be used for manipulating "smart humans"[10] , like robots, and controlling their movements. This enhances the meaning of gesture-based interaction and leads the developers to make systems which will be helpful to human's reality and everyday life and to create "smart environments" with robots and interaction techniques where the user shall feel comfortable with and the master of the communication.

- Context Awareness

Context aware systems generally track users and through that offer a large range of information on demand. Context awareness may be used as a mean to provide navigation in 3D space. Another form in which context awareness appears in literature is the progressive information visualization. When the user is far away from the system then the information displayed is more like an overview and when the user come closer the system displays more details of information. [11]

- Facial Expressions Recognition

Facial expressions are the face changes in response to a person's internal emotional states, intentions, or social communications. Computer recognition of facial expressions has many important applications such as intelligent human computer interaction, computer

animation, and awareness systems.[4]. The expressions of the face are usually being used in game agents to provide emotions to the players and to express their companionship [28]. But, “Facial expressions have long been considered the “universal language of emotion”” [68]. Through researches and psychophysical techniques on the 6 basic facial expressions (happy, surprise, fear, disgust, anger and sad) it observed cultural diversity on facial expressions.

- Interaction with mobile devices

Nowadays, many people have smart phones and it is reasonable to expect that they carry them even when they are playing games. Modern smart phones contain sophisticated sensors to monitor three-dimensional movement of the device. These sensors permit devices to recognize motion gestures deliberate movements of the device by end-users to invoke commands. Researchers have proposed the use of motion gestures for a variety of input tasks: for example, to navigate maps or images, to input text , to control a cursor, and to verify user identity [20]. Almost all the smartphones have accelerometer, gyroscope, microphone, etc. When the user is moving, the smartphone sensors can produce information related to the movement. Smartphones can help to improve the accuracy and sensitivity of Kinect and potentially lessen its limitations. [19]. Users generally like gestures as a way to use their mobile phones to intuitively interact with other devices.[21]

2.2 Existing Approaches for Virtual Humans

An agent is a system that operates independently and rationally, seeking to achieve its goals by interacting with its environment. It has goals and beliefs, and executes actions based on those goals and beliefs [22]**.** In computer science, a system agent is a computer program that acts for a user or other program in a relationship of agency, which derives from the Latin *agere* [78] (to do): an agreement to act on one's behalf. Such "action on behalf of" implies the authority to decide which, if any, action is appropriate

In artificial intelligence, an intelligent agent (IA) is an autonomous entity which observes through sensors and acts upon an environment using actuators (i.e. it is an agent) and directs its activity towards achieving goals (i.e. it is rational). Intelligent agents may also learn or use knowledge to achieve their goals. An autonomous agent is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future.

Intelligent agents (IA) have been defined many different ways. According to Kasabov [23] IA systems should exhibit the following characteristics:

- accommodate new problem solving rules incrementally
- adapt online and in real time
- be able to analyze itself in terms of behavior, error and success.
- learn and improve through interaction with the environment (embodiment)
- learn quickly from large amounts of data
- have memory-based exemplar storage and retrieval capacities
- have parameters to represent short and long term memory, age, forgetting, etc.

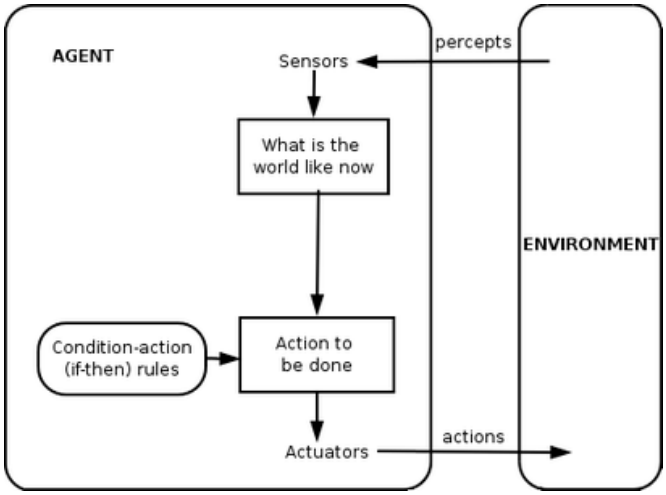


Fig. 3: Simple reflex agent

Virtual humans are software artifacts that look like, act like, and interact with humans but exist in virtual environments. Autonomous agents support face-to-face communication in a variety of roles can enrich interactive virtual worlds. Autonomous virtual agents are essential such as *inhabited virtual environments,* like airports, museums and other public spaces, where their role is assistive and informative most of the times. Furthermore, virtual humans may be applied as medical assistants for remote care of patients [25] [57]. They can also be used as intelligent computer-generated substitutes of real persons by acting instead or behalf of them on the network. They have the voice and appearance of the real persons and communicate with people or other virtual objects/humans. The most important reason for creating virtual humanoid agents is to be able to use them in virtual scenes as believable people and to take a role of a real interlocutor.

2.2.1 Personalization of Human – Agent Interaction

James Doman [58] states that "Personalization technology enables the dynamic insertion, customization or suggestion of content in any format that is relevant to the individual user, based on the user's implicit behavior and preferences, and explicitly given details". Personalization involves a process of gathering user-information during interaction with the user, which is then used to provide appropriate assistance or services, tailor-made to the user's needs. Personalization is motivated by the recognition that a user has needs, and meeting them successfully is likely to lead to a satisfying relationship with him.

In the domain of multi-user and agent-oriented information systems, personalized information systems aim to give specific and customized responses to individual user requests. In addition to the ability to analyze user needs and to retrieve, understand and act on distributed data that is offered by any agent-oriented system, multi-agent systems also offer interesting possibilities for interaction, particularly with regard to information sharing and task coordination.

The main functions of an information agent have been summed up by Petit Roze [40]: information acquisition and management, information synthesis and presentation, and intelligent user assistance. Agents possess several interesting characteristics in terms of information system design, including:

- proactivity, which allows the triggering of actions that have not been explicitly requested, meaning, for example, that a warning can be activated if an agent receives information that it deems useful for some users;
- uncertainty management, which is a key feature in Artificial Intelligence that allows agents to infer from their current incomplete knowledge and past experiences, making assumptions to compensate for lack of knowledge and/or learning from previous user transactions;
- autonomy, which allows agents to deal with distributed data and knowledge or processing resources; and
- social abilities, which allow agents in multi-agent systems to perform tasks requiring interaction between distributed entities, including knowledge sharing and task coordination

The agents' ability to both analyze user needs and to retrieve, understand and act on distributed data and knowledge about users is fundamental to agent-oriented information system design. However, further agency of the system is required in order for information to be exchanged cooperatively. MAPIS, the Multi-Agent Personalized Information System

provides personalized access to a set of information, by interacting with both users and information sources (Fig. 4). It employs four types of roles and four agents for each role: assistive agents mediate between the users and the system, search agents look for data at its sources, profile agents manage the user model and solver agents coordinate the information retrieval and personalization processes and integrate the data to generate an appropriate solution.

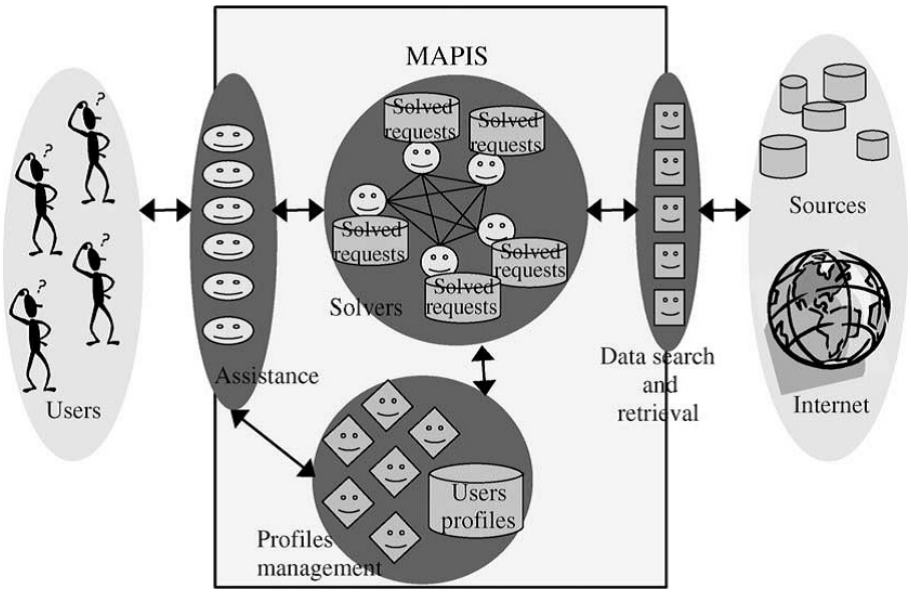


Fig. 4: General Description of MAPIS

The technological evolution of recent years has allowed the design of so-called pervasive systems, which are capable to offer, as transparently as possible, advanced services to the user. Personal mobile devices equipped with wireless and localization technologies can be employed to develop pervasive and context-aware systems [1] with the aim to provide people useful information in relation with the user profile and with the environment within the user is in. New services are proposed and new ways of service provision are studied according to the features and capabilities of mobile devices [36,37]. This common interest is justified by the wide diffusion of such devices, that are more or less in everyone's pocket and that can be used almost everywhere and anytime. Santangelo proposes a PDA-based personal shopper assistant, which is a system that aims at providing users with user-friendly guide during their shopping time. It supplies skills of a human-like shopper assistant and stores user preferences to suggest shops of possible interest at any

time. The proposed architecture is based on a client-server paradigm. It is accessible from small handheld devices like cell phones and PDAs that contain sufficient processing power to handle a variety of tasks (Fig. 5).

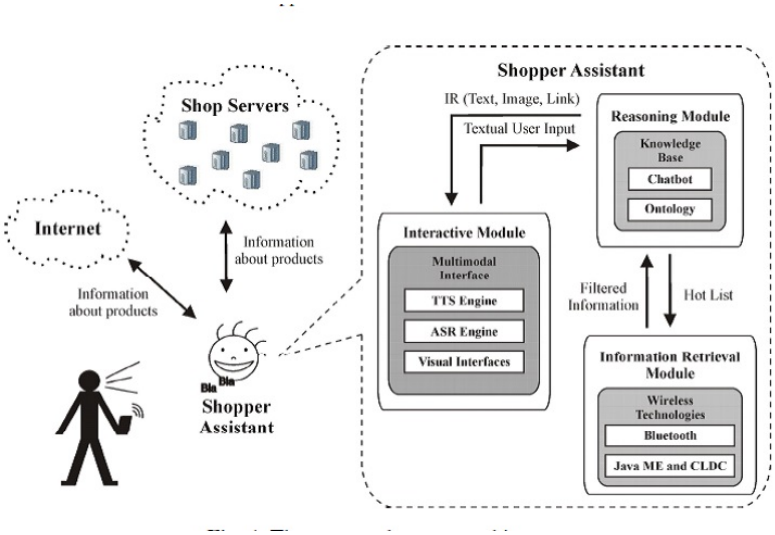


Fig. 5: Virtual Shopper Customer Assistant architecture

Interface agents have become a technology widely used to provide personalized assistance to users with their computer-based tasks. Most interface agents achieve personalization by learning a user's preferences in a given application domain and assisting him accordingly. In order to personalize the interaction with users, interface agents should also learn how to best interact with each user and how to provide them assistance of the right sort at the right time. To fulfill this goal, an interface agent has to discover when the user wants a suggestion to solve a problem or deal with a given situation, when he requires only a warning about it and when he does not need any assistance at all. Schiaffino [49] approaches personalization in a different way, on how to personalize the interaction between interface agents and users in a mixed-initiative interaction context [Personalizing user-agent interaction] and proposes a learning algorithm, named WoS, to tackle this problem. The algorithm is based on the observation of a user's actions and on a user's reactions to the agent's assistance actions. The WoS algorithm enables an interface agent to adapt its behavior and its interaction with a user to the user's assistance requirements in each particular context. From this work as results came several issues that agent developers have to address in order to improve interaction via personalization between agents and users, as:

- Discovering the type of assistant each user wants.

- Considering the particular assistance requirements users have in different contexts.
- Analyzing users' tolerance to agents' errors.
- Discovering when (context awareness), and when not, to interrupt the user.
- Providing the means to provide simple (but useful) explicit user feedback.
- Providing the means to capture as much implicit feedback as possible.
- Discovering how much control the user wants to delegate to the agent.
- Providing the means to control and inspect agent behavior.

Kasap and Thalmann present [65] a state-of the-art survey on virtual humans, mentioning the use of intelligent decision technologies in order to build virtual human architectures and considering various aspects such as autonomy, interaction and personification. Each of these aspects comes to prominence in different applications. In their survey, the authors review the autonomous behavior of virtual characters considering both the internal state of the virtual human and the state of the virtual environment. They present the importance of interaction capabilities of virtual humans by giving examples of Embodied Conversational Agents (ECAs) and mentioning about different components of interaction such as facial expressions, gestures and dialogue. Finally, another important factor of virtual characters called personification is considered.

2.2.2 Training Systems including Agents

Virtual environments can be incredible training tools if used properly and used for the correct training application. Training systems of the future need to simulate all aspects of a virtual world, from physics of scene objects to realistic human behavior. So, projects, like Virtual Humans [24,75] at the Institute of Creative Technology (ICT) are concentrating on high fidelity embodied agents that are integrated into these environments. These agents are aimed to provide a social and human focus to training and serve with roles that support interactive face-to-face interaction, delivering a powerful mechanism for training interpersonal skills and experiential education. These interpersonal skills require a vast knowledge of the various aspects of human behavior that is difficult to formalize and appropriately display. So as to effectively perform this mission, building virtual humans that have the capability to interact with trainees on this interpersonal level is required. By incorporating this set of human behavior with virtual characters, virtual worlds can be made applicable to a wide range of training tasks that currently require labor-intensive live exercises, role playing, or are taught non-experientially. This potential depends on the

success in creating characters that convey three main characteristics, as mentioned in [70]. Firstly, the virtual characters should be *believable* and provide a sufficient illusion of human-like behavior. Secondly, they should be *responsive* in order to react naturally to the human user and to the events coming from the environment. Finally, they should be *interpretable* so the users can without doubt understand their responses to situations, their dynamic cognitive and emotional state.

To master complex tasks people need help on facing a wide range of situations. They also need a mentor that can demonstrate procedures, answer questions, and monitor their performance, and they may need teammates if their task requires multiple people. Since it is often impractical to provide such training on real equipment, the use of virtual reality has been explored instead and autonomous animated virtual agents have taken the role of mentor and teammates . Such an indicative example is Steve(Soar Training Expert for Virtual Environments, Fig. 6) at [50], which integrates from three research areas: intelligent tutoring systems, computer graphics and agent architecture. Steve can demonstrate how to perform tasks and can also monitor students while they practice tasks, providing assistance when needed.

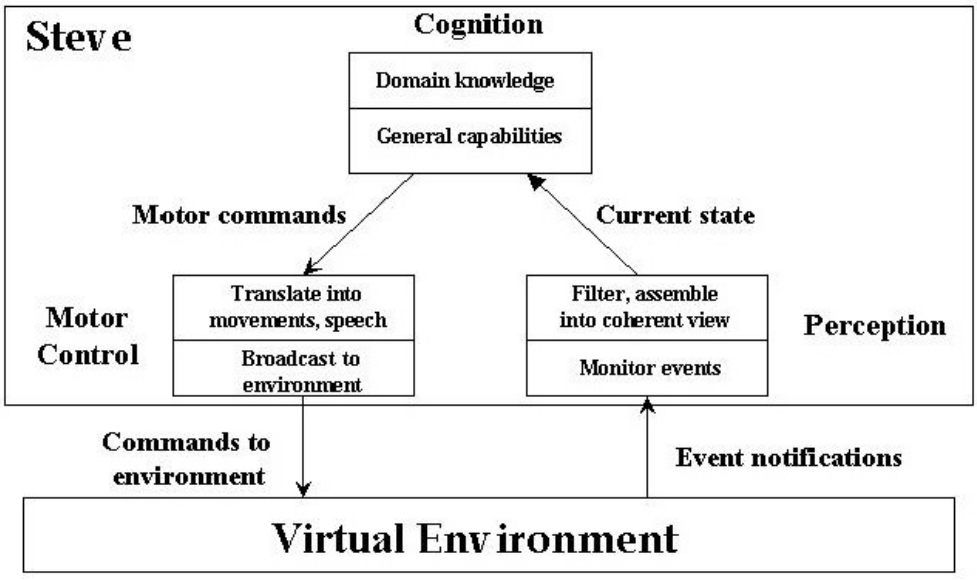


Fig. 6: Steve's three main modules (perception, cognition, and motor control) and the types of information they send and receive

Team tasks are ubiquitous in today's society. So, training systems have been made and for team training as well. Rickel and Johnson at [50] are describing a virtual world with virtual humans and virtual reality distribution, where students, instructors and virtual humans share a three-dimensional, interactive, simulated mock-up of their work

environment and practice together in realistic situations. The virtual humans can serve as instructors for individual students, and they can substitute for missing team members, allowing students to practice team tasks when some or all human instructors and teammates are unavailable. This learning system, which is an extension of Steve agent previously used for one-to-one tutoring, induct students to learn their individual role in the team as well as how to coordinate their actions with their teammates.

Finally, the work proposed in [44] describes an augmented reality case study where a virtual teacher trains a novice user to handle complex machines. Real machineries and surroundings are used in order to further enhance the realism of the scene.

2.2.3 Embodied Conversational and Emotional Agents

Embodied conversational agents (Fig. 7) [38] are a form of intelligent user interface. Graphically embodied agents aim to unite gesture, facial expression and speech to enable face-to-face communication with users, providing a powerful means of human-computer interaction.

Fig. 7: Conversational Agents

Embodied Conversational Agent (ECA) is the user interface metaphor that allows to naturally communicating information during human-computer interaction in synergic modality dimensions, including voice, gesture, emotion, text, etc. Due to its anthropological representation and the ability to express humanlike behavior, ECAs are becoming popular interface front-ends for dialog and conversational applications. Natural interaction, such as speech, involves both verbal and non-verbal communication acts. Kunc at [39] introduces

the ECAF (Embodied Conversational Agent Facade), an ECA authoring language which focuses on the needs of developers to build multimodal applications with ECA-based interfaces. (Fig. 8) The proliferation of animated characters can be enhanced by the existence of effective programming languages and architectures supporting real-time generation of behavior based on the higher-level expressive concepts.

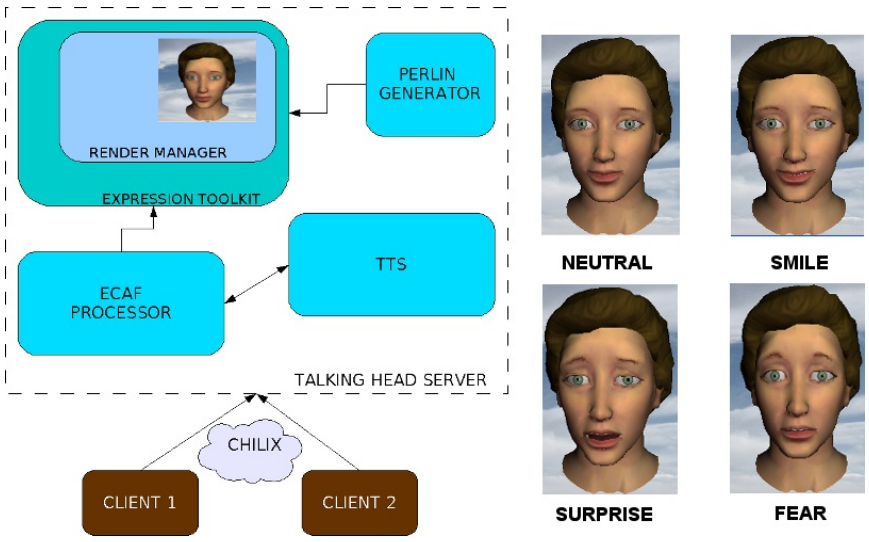


Fig. 8: System architecture and examples of talking head expressions

Because many sophisticated computer science research areas are required to create virtual humans, in addition to serving as a guide, a virtual human can itself serve as an exhibit of technology. Consolidating this idea InterFaces project has created virtual museum guides (Fig. 9) that are in use at the Museum of Science in Boston. Their goal was to provide information but also engage the visitor in an interactive exchange that can lead to deeper understanding and promote excitement about the museum content. The characters use natural language interaction and have near photo-real appearance to enhance user experience and present reports from museum staff on visitor reaction. [30]

Fig. 9: Visitors engaging with Ada and Grace, guides at the Museum of Science in Boston

Virtual Humans that can interact with users in immersive virtual worlds play many important roles as they are used for education, training and entertainment. Perhaps the greatest challenge in creating virtual humans for interactive experiences is supporting face-to-face communication among people and virtual humans. Virtual worlds, because of the fact that there will usually be multiple real and virtual people, require support of multi-party conversations, including the ability to reason about the active participants in a conversation as well as who else might be listening or uninformed of what happens in a conversation.

Traum et al. [52] introduces a candidate model that integrates and extends prior work on spoken dialogue and ECAs, providing an expansive establishment for multi-party dialogues in immersive virtual worlds, drawing on prior models of collaborative dialogue from computational linguistics, as well as work on ECAs and the social psychology literature on the nonverbal signals that accompany human speech.

The avatar is a type of embodied agent that has received much airplay but not in-depth research attention in the agent community. Avatars typically serve as presence displays, rather than actually contributing to the experience of having a face-to-face conversation. BodyChat (Fig. 10) [56] is a system that allows users to communicate via text while avatars automatically animate attention, salutation, turn taking, back-channel feedback and facial expressions. On this system evaluation results showed that users found an avatar with autonomous conversational behaviors to be more natural than avatars whose behaviors they controlled and in aid of this that autonomous avatars with communicative behaviors (Fig. 11, Fig. 12, Fig. 13) provided a greater sense of user control.

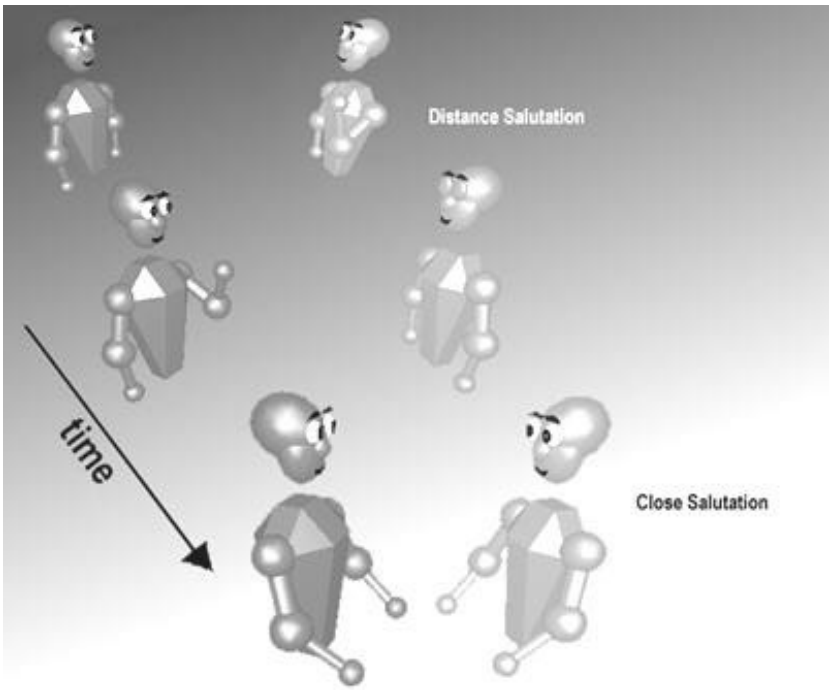


Fig. 10: Avatars A and B exchange Distance Salutations when the system registers them as conversational partners. When they get within a conversational range, Close Salutations are exchanged.

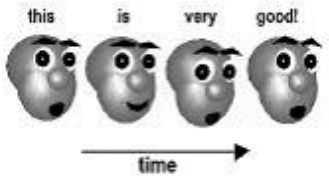


Fig. 11: Some words are accompanied with a special facial expression.

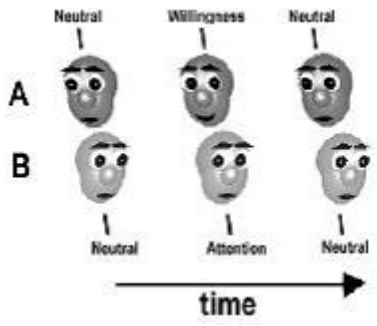


Fig. 12: The sequence of glances when user A clicks in avatar B to express willingness to chat while user B is not available.

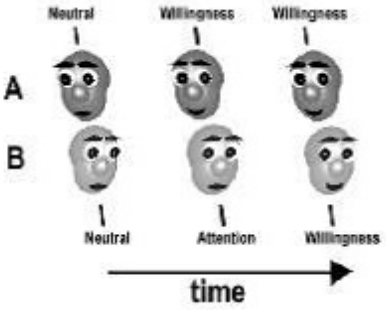


Fig. 13: The sequence of glances when user A clicks on avatar B to express willingness to chat and user B is available.

Besides modeling the "mind" and creating intelligent communication behavior on the encoding side, which is an active field of research in artificial intelligence, the visual representation of a character including its perceivable behavior from a decoding perspective, such as facial expressions and gestures, belongs to the domain of computer graphics and likewise implicates many open issues concerning natural communication. Therefore, Jung et al [64] give a comprehensive overview on how to go from communication

models to actual animation and rendering. They focus on the visualization component of multimodal dialog systems and mainly deal with the graphical realization of the embodied agent, including its nonverbal output.

2.2.4 Virtual Human Engines and Toolkits

Virtual humans can be powerful tools in a wide range of areas. While virtual humans are proven tools for training, education and research, they are far from realizing their full potential. Realizing their full potential requires compelling characters that can engage users in meaningful and realistic social interactions, and an ability to develop these characters effectively and efficiently. First, in order for virtual humans to be effective, they need to exhibit a range of capabilities, simulating those of real humans. Second, it is important that these abilities not work only in isolation, they also need to be integrated into a larger system (and further, into systems of systems), where they can inform, influence and strengthen each other. Third, even with the appropriate knowledge and resources available, virtual humans can still be costly to develop. Furthermore, certain principles may only be understood in a narrow context and can be difficult to generalize across multiple domains. This limits the ability to re-use knowledge and assets, often resulting in the need to start new characters or systems from scratch.

To address these challenges ICT Virtual Human Toolkit [24] was designed to support researchers with the creation of embodied conversational agents (Fig. 14). It offers a collection of modules, tools, libraries, a framework and open architecture that integrates these components. Apart from virtual agents the Toolkit services full coverage of subareas such as speech recognition, audio-visual sensing, natural language processing, dialogue management, nonverbal behavior generation and realization, text to speech and rendering.

Fig. 14: Rachel (left) and Brad (right), Agents from ICT VHT

The generation of expressive 3D characters requires a series of stages, including the generation of a character model, specifying a skeleton for that model, deforming the model according to the movement of the skeleton, applying motion and control algorithms under a framework, and finally instructing the character to perform. While many high quality assets such as humanoid models or motion capture data can be willingly and inexpensively gained, the integration of such assets into a working 3D character is not automated and requires expert interference. The difficulty of animating 3D virtual characters presents a barrier for the end user, who cannot easily control a 3D character without the assistance of specialists, regardless of the broad availability of the models, assets and simulation environments. To address this problem a system that allows the integration of high-fidelity humanoid 3D models into simulation was created by Feng at [26]. The system's pipeline relies upon two main processes: 1) an automated skeleton matching process and 2) a retargeting process that can transfer motion sets onto a new character without user interference (Fig. 15).

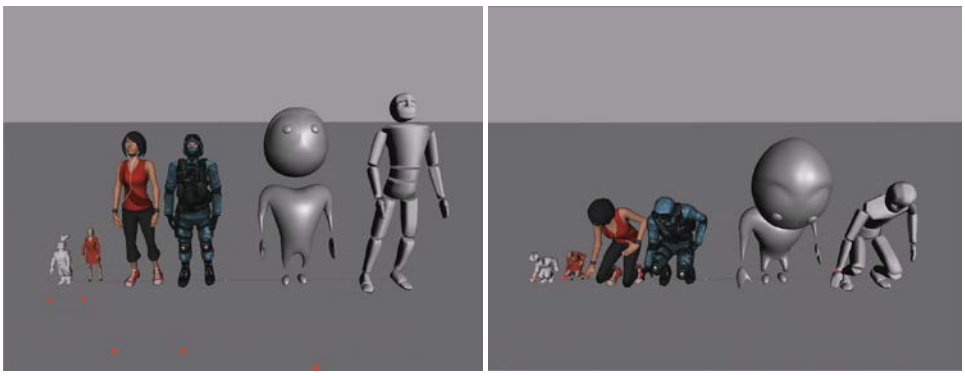


Fig. 15: A set of characters from many different sources are automatically retargeted and registered into our system

Another powerful animation engine for developing applications with embodied animated agents is Maxine [34]. The engine allows management of scenes and virtual characters and focuses on multimodal and emotional interaction in order to establish more effective communication with the user. Particular emphasis is given on capturing the user's emotions through images, as well as on producing the virtual agent's emotions through facial expressions and voice adjustment. (Fig. 16, Fig. 17) In Maxine the virtual agent is provided with the following distinctive features: it supports interaction with the user through different controls as text, voice, peripherals, it gathers further information on the user and his/her surroundings, it supports voice communication in natural language and it has own emotional state which may vary depending on the relationship with the user. (Fig. 18)

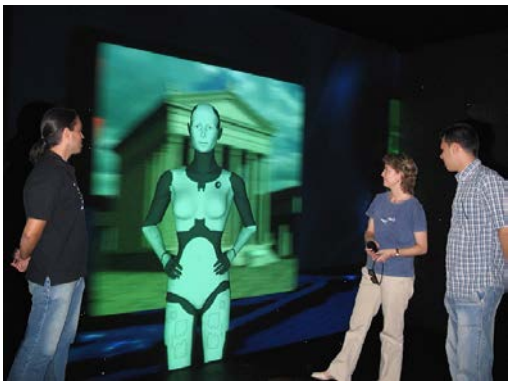


Fig. 16: Maxine's presentation in immersive environment

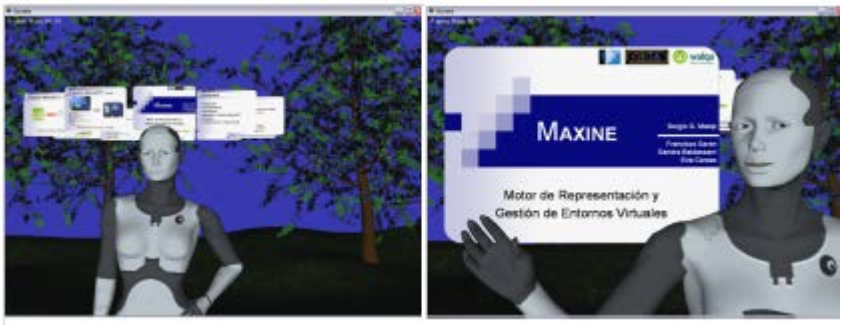


Fig. 17: Screenshot from virtual presentation

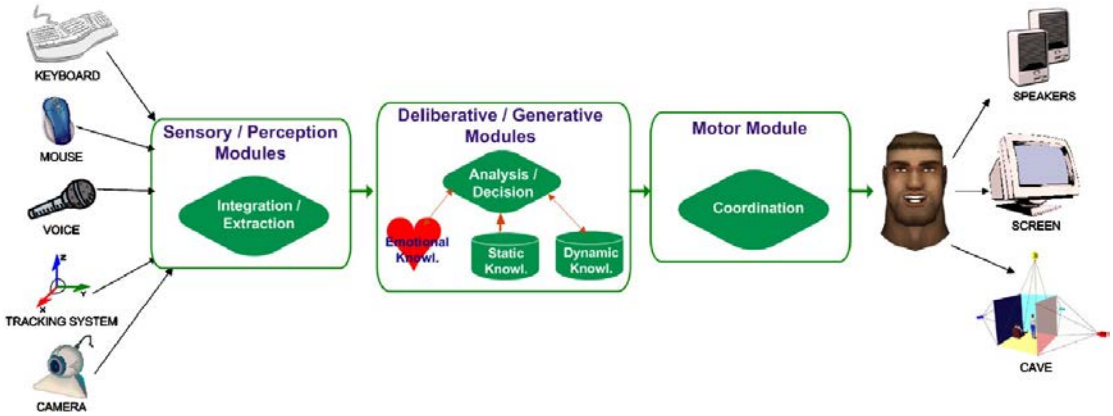


Fig. 18: Maxine's architecture

The Behavior Expression Animation Toolkit (BEAT) [35] is a another system that allows animators to input typed text that they wish to be spoken by an animated human figure (Fig. 19) and to obtain as output appropriate and synchronized nonverbal behaviors and synthesized speech in a form that can be sent to a number of different animation systems. It uses linguistic and contextual information contained in the text to control the

movements of the hands, arms and face, and the intonation of the voice. The mapping from text to facial, accentuation and body gestures is contained in a set of rules derived from the state of the art in nonverbal conversational behavior research. The toolkit is extensible, so that new rules can be added from animators concerning personality, movement characteristics and other features that are comprehended in the final animation. The BEAT toolkit is the first of a new generation (the *beat generation*) of animation tool that extracts actual linguistic and contextual information from text in order to suggest appropriate gestures, eye gaze, and other nonverbal behaviors, and to synchronize those behaviors to one another.

Fig. 19: "I don't know if this is a good thing or a bad thing", virtual figure from BEAT

One of the main assets of virtual humans is that they can support ordinary language and be able to communicate with users in a natural way by listening to what they say and react suitably. NPCEditor is a tool at [32] that allows easy construction, maintenance and run-time deployment of language processing capabilities for virtual characters. NPCEditor supports design and development of a natural language understanding (NLU) component of a virtual human.(Fig. 20) The component accepts an input from the user of the virtual human system (generally in text string) and returns an appropriate response.

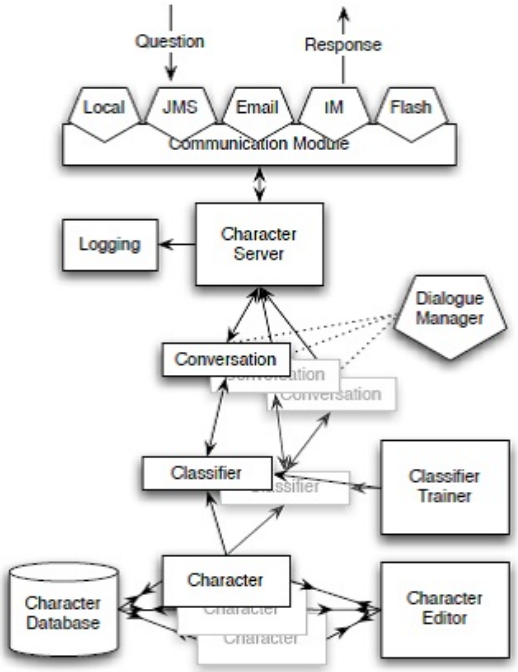


Fig. 20: NPCEditor system design

Virtual character animations are an integral part of virtual agents by offering expressiveness and realism. In this direction, Chaudhuri et al. [43] present an efficient approach for character animation using OpenSceneGraph [77], in which the animation is adapted according to the user's point of view. The key idea of their work involves adapting animations provided by animators with regard to the user's perspective, so as to offer a more immersive and realistic visualization of the virtual characters. The authors' implementation based on dual quaternions and user tracking relies on computer vision techniques, making their approach suitable for domains such as virtual and augmented reality or in ambient intelligence environments. Furthermore, Papagiannakis [45] proposes applying Geometric Algebra so as to improve the performance of real-time animation blending in comparison to dual quaternion implementations.

Gillies and Bernhard [66] provide an overview of the state of the art in character engines and propose taxonomy of the features that are commonly found in them. This taxonomy can be used as a tool for the comparison and evaluation of different engines. A demonstration of this tool through the comparison of three engines (Cal3D, Piavca and HALCA) is showed. A brief discussion of some other popular engines is also given.

2.2.5 Adaptive Behaviors in Multimodal Human-Agent Interaction

Multimodal interaction seems easy in everyday life, but a closer look exposes that such interaction is indeed complex and contains a variety of coordination levels, from high-level linguistic exchanges to low-level pairings of temporary body movements both within an agent and across multiple interactive agents. If a better understanding of how these multimodal behaviors are coordinated exists then insightful principles to guide the development of intelligent multimodal interfaces can be provided. In light of this, Zhang at [31] proposes a real-time multimodal human-agent interaction system in which human contributors interact with a virtual agent in a virtual environment (Fig. 21). This platform tolerates the virtual agent to keep track of the user's gaze and hand movements in real time and according to this adjust its own behavior. This approach for studying multimodal human-avatar interaction has three main goals while building and using such a framework: 1) to test and evaluate interactive behavioral models in human-agent interaction, 2) to develop, test and evaluate cognitive models that can emulate those patterns and 3) to develop, test and design new human agent multimodal interfaces which include the virtual agent, the control policy and real time adaptive human-like behaviors.

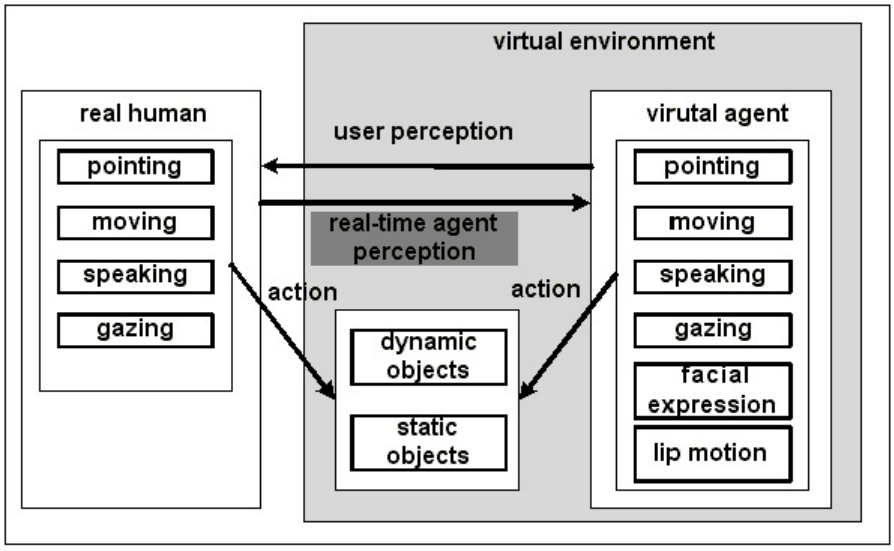


Fig. 21: Overview of the research platform to investigate multimodal human-avatar interactions. A real person and a virtual human interact with each other in a virtual environment. The system controls the actions of the virtual person and measures the behavioral response actions accordingly.

As defined by Mark W. Bell, a virtual world is a "synchronous, persistent network of people, represented as avatars, facilitated by networked computers." [78,59]. Virtual worlds have become largely synonymous with interactive 3D virtual environments. He also states that a virtual world is a composition of architectural metaphors and computing entities. The

architectural metaphors are useful to provide a sense of place and, if multi-user, a sense of awareness of others. An agent is a system that operates independently and rationally, seeking to achieve its goals by interacting with its environment. It has goals and beliefs, and executes actions based on those goals and beliefs [22]. Maher et al. [54] develop an agent model of 3D virtual worlds that assumes a persistent object-oriented representation of the world and give each object agency beyond 3D object representation. Furthermore, the authors propose a way to extend the concept of virtual worlds from preprogrammed interactive 3D models to places with objects that respond to their use by reasoning about the environment and then modifying the environment. Each object in the world is an agent element so that the world is a society of agents. Each agent element can sense and respond to the current state of the world, as illustrated in the figure (Fig. 22) below.

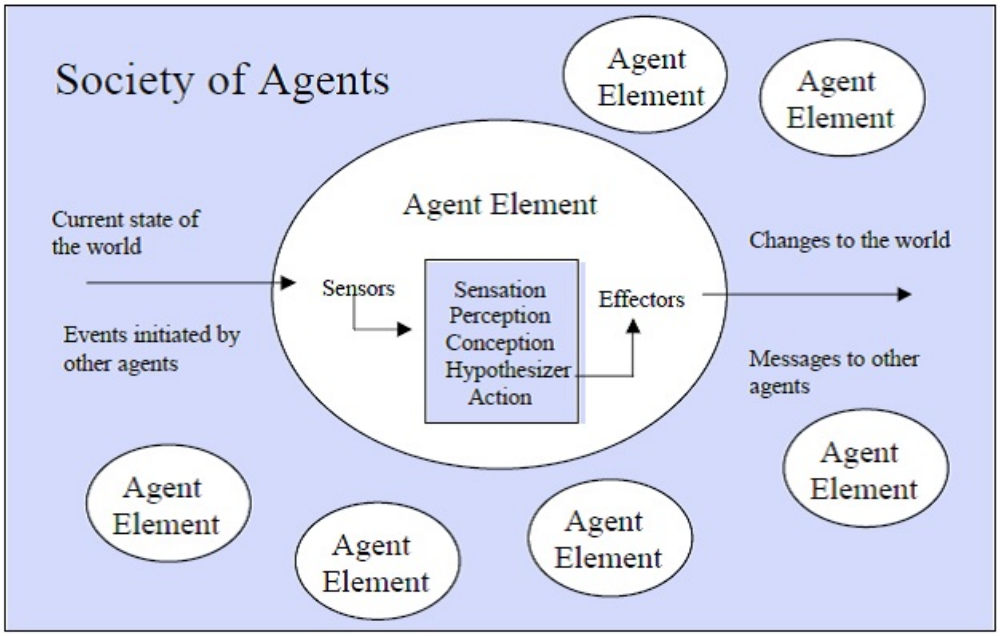


Fig. 22: Each agent element can sense and respond to the current state of the world

This response can result in a dynamic world that configures and reconfigures itself as needed. The agent model here assumes different layers of reasoning that provide flexibility in the behavior of the agent. By sensation, perception, conception, hypothesizing and action separation, intelligent objects are developed to reason and act on different levels of abstraction. This effectively defines an intelligent world as a society of intelligent agents.

An Embodied Conversational Agent, by Cavalluzzi et al. [42], represents a creative expression of a natural and appealing interface between users and services of smart

environments (Fig. 23). This work proposes the personalization of conversations with an embodied intelligent agent in public spaces, where interaction may be performed using a public touch screen or a personal device. The agent communication is adapted to the situation at both content and presentation levels, by generating an appropriate combination of verbal and non-verbal agent behaviors, and when needed, to increase the sense of social relation with the user by showing an empathic attitude.

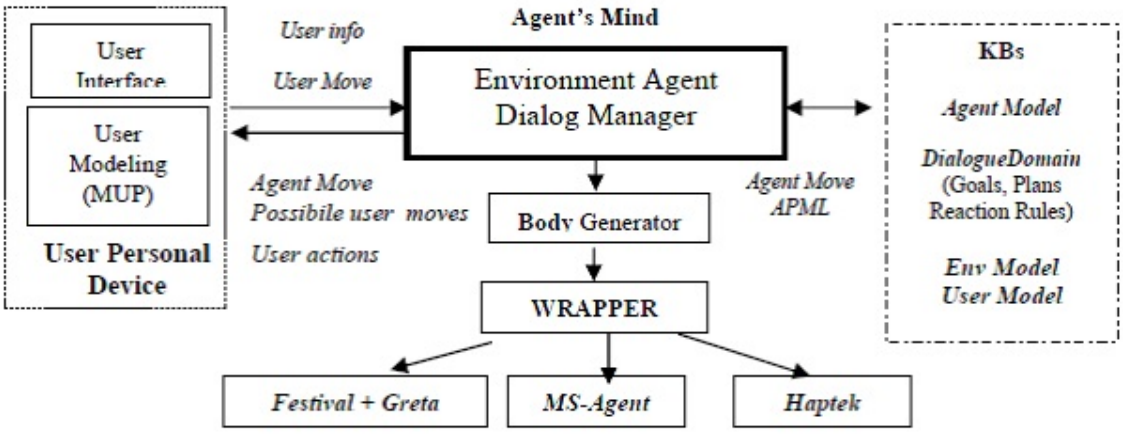


Fig. 23:Functional components of our conversational system

2.2.6 Human -Agent Interaction in Ambient Intelligence Environments

The last decades more and more virtual humans are being used in Ambient Intelligence environments in order to help, assist and guide humans to understand the systems and the ubiquitous technology surrounding them. An issue of the existence of virtual agents in ambient intelligence is that they should embody themselves in the environment and naturally communicate with the users. Due to their humanoid appearance, the human-agent interaction should be as physical as the human-human interaction is. So, the ambient intelligence environments that contain virtual agents in their scenes provide the appropriate interaction techniques for the best collaboration with the users. These techniques include speech recognition and speech synthesis for linguistic interaction as well as conversation with the user, because a considerable part of human communication is based on speech. Therefore, a believable virtual humanoid environment with user interaction should include speech recognition. Furthermore, head movement, gaze detection, body gestures, user tracking are additional interaction techniques applied for

adaptive verbal and nonverbal behavior of the virtual agent. Body and hand gestures are additionally used for the users' interaction with the system and the agent as well as mobile interaction for remote control of the environment and the assistant. Finally, the traditional interaction techniques as mouse and touch events and keyboard are used too. Several examples from literature which are relative to human-agent interaction in ambient intelligence environments are presented below.

Santangelo [36] proposes a human-like PDA based personal shopper assistant, which is able to understand the user needs via spoken language interaction and subsequently stores the preferences of the potential customer. The interaction is given through automatic speech recognition and speech synthesis technologies (vocal interaction). in addition, localization is allowed through wireless tools and the personal assistant makes use of a reasoning engine in order to make inference, store and retrieve user preferences. Bluetooth devices detection and communication technologies allow the personal assistant to alert the user when a shop is close to his current physical position filtering products information according to customer preferences parameters.

COHIBIT (**CO**nversational **H**elpers in an **I**mmersive exhi**BI**t with a **T**angible interface) [41] is an edutainment exhibit for theme parks in an ambient intelligence environment. The visitors can use instrumented 3D puzzle pieces to assemble a car (Fig. 24). The key idea of this edutainment framework is that all actions of a visitor are tracked and commented by two life-like guides, which observe, follow and understand the user actions and provide guidance and motivation for them. A tangible (via the graspable car pieces), multimodal (via the coordinated speech, gestures and body language of the virtual character team) and immersive (via the large-size projection of the life-like characters) experience is given to the visitor(s) through the mixed reality environment. The exhibit requires interaction modalities to allow the design of a natural, modest and robust interaction with the intelligent virtual agents. For this purpose, Radio Frequency Identification Devices (RFID) are used to determine the position and orientation of the car pieces wirelessly in the instrumented environment.

Fig. 24: Examples of visitor actions and corresponding character behavior

Interaction is defined as mutual or reciprocal action or influence. In real world everything and everybody interacts with everyone. Several methods have been developed for interactions inside a homogenous Virtual Environment, and between an application and its user. Balcisoy's et al approach [51] is that using mixed environments and virtual humans as a mediator between a real environment and a virtual environment, a natural way of interaction between human and machine can be achieved. A virtual human can act very human like in terms of animation, ergonomics, and perform very precise handling of a virtual object at the same time. The interaction with the virtual human is defined as triggering some meaningful reactions in the environment in response actions as body gestures or verbal output from a participant. The techniques that the authors are using for interaction are firstly direct manipulation of objects in a mixed reality in order that the participants interact

with the world by becoming a part of the system and not through a GUI or a device. With the second way of interaction, users perform precise operation on real and virtual objects in a mixed environment using a semi-autonomous virtual human (Fig. 25), with the agent to play the role of mediator between real and virtual world.

Fig. 25: Checkers play between real and virtual human

The Jacob (Fig. 26)project [62] involves the construction of a 3D virtual environment where an animated human-like agent called Jacob gives instruction and assistance for tasks that the user has to execute in a virtual world. The user interacts with Jacob by performing actions as well as by using natural language. The visualization of Jacob's body has been created to comply with the H-Anim standard [76], which is a standard for describing a humanoid body in terms of joints and segments. The use of this standard makes it relatively easy to plug in a different body for Jacob.

Fig. 26: Jacob's prototype

Providing the user with an intuitive interface to control the avatar's motion is difficult because the character's motion is high dimensional and most of the available input devices are not. Input from devices such as mice and joysticks typically indicate a position, velocity or behavior. Control of individual degrees of freedom is not possible for interactive environments unless users can use their own body to act out or mimic the motion. At [55] the authors present three interface techniques for controlling avatar motion: the user selects from a set of available choices, sketches a path through an environment, or acts out a desired motion in front of a video camera (Fig. 27).

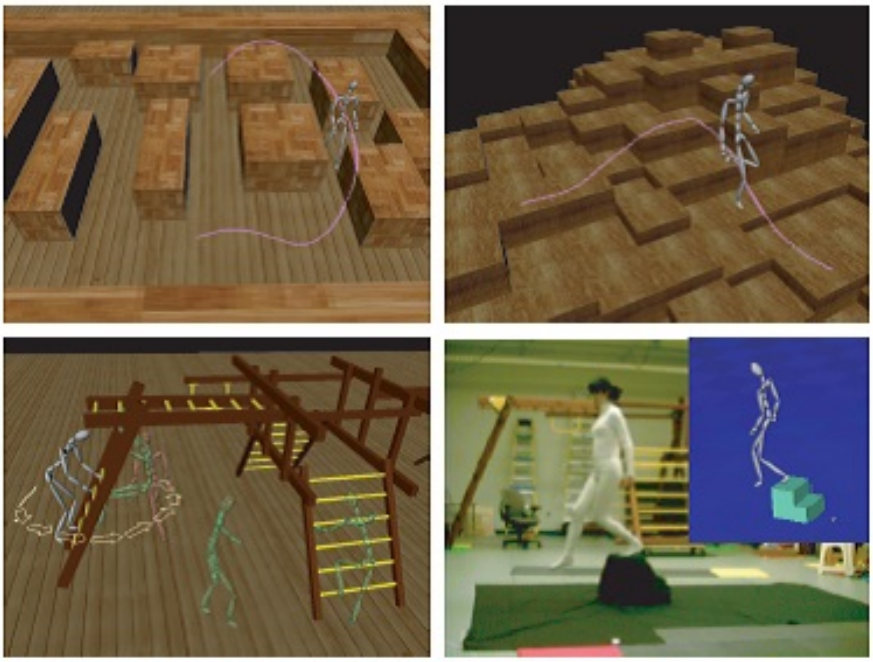


Fig. 27: Real-time avatar control in the system. (Top) The user controls the avatar's motion using sketched paths in maze and rough terrain environments. (Bottom left) The user selects from a number of choices in a playground environment. (Bottom Right) The user is controlling the avatar by performing a motion in front of a camera.

Egges et al [67] present a simple and robust Mixed Reality (MR) framework that supports real-time interaction with Virtual Humans in real and virtual environments under consistent illumination. They focus on three crucial parts of the framework: interaction, animation and global illumination of virtual humans for an integrated and enhanced presence. The system includes speech recognition, speech synthesis, interaction, emotion and personality simulation, real-time face and body animation and synthesis, real-time camera tracking for AR and real-time virtual human Precomputed Radiance Transfer [Fig. 28] rendering, and is able to run at acceptable speeds for real-time (20-30fps) on a normal PC.

They finally provide some different scenarios where the system is running in MR applications.

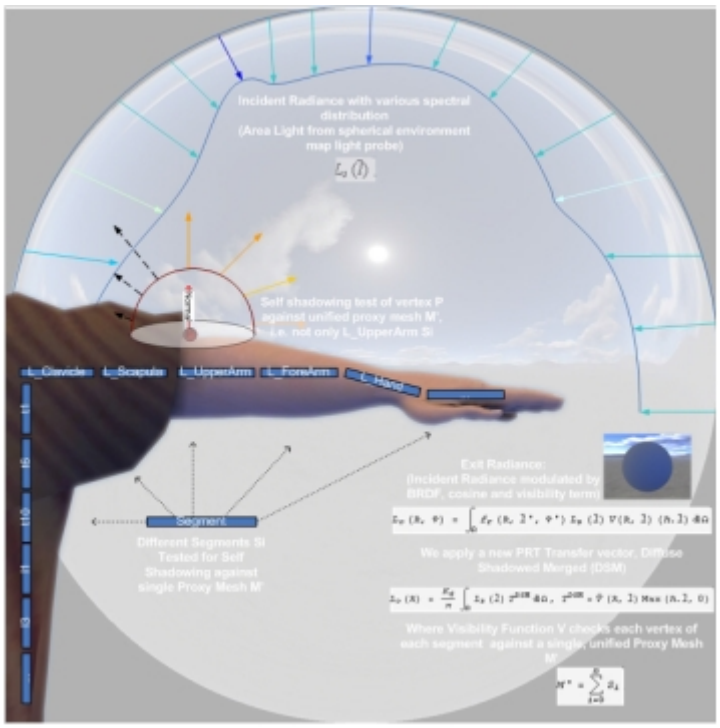


Fig. 28: Schematic view of dPRT illumination model

2.2.7 Standalone Applications with Virtual Agents

Apart from frameworks, assistive tools and engines, literature contains additional examples of the applications of virtual agents that act as standalone applications. These agents aim to interact and offer diverse types of information to their users. Several indicative examples are presented in detail below.

- **AmiQuin [33]**

  AmiQuin is a virtual mannequin which leverages an Ambient Intelligence system within a shopping environment. It is designed to replace a traditional shop window mannequin in order to enhance a customer's shopping experience by reaching to the customer's presence and presenting personalized information.

- **Archeoguide [53]**

  Archeoguide offers personalized augmented reality tours of archeological sites. It uses outdoor tracking, mobile computing, 3D visualization and augmented reality

techniques to enhance information presentation, reconstruct ruined sites and simulate ancient life.

- **Speak out and annoy –the vision kiosk [61]**

  Vision kiosk's design is centered on the use of an animated talking head as the focus of user interaction. The avatar turns and watches approaching clients and when the clients get close enough and appear interested in the kiosk it speaks a greeting. As the user navigates the information on the kiosk, the avatar provides assistance with useful information and sharp wordplay. The session ends when the user leaves from the kiosk and so the avatar goodbyes him/her.

- **Tele Health Apps [27]**

  MultiSense is a software package that combines audio and video information to recognize clinically relevant nonverbal behavior. This module acts as a pluggable component that can inform a variety of potential health applications. Gratch et al. apply this approach on two methods, one using a virtual health agent and the employing using telemedicine, for performing PTSD (Post-Traumatic Stress Disorder) and depression screening for U.S.

  SimSensei (Fig. 29) explores the advisability of virtual health agents for mental health screening. Engages users in a structural interview using natural language and nonverbal sensing with the aim of identifying risk issues associated with depression or PTSD.

  TeleCoach (Fig. 29*)* explores the use of user state sensing to inform telemedicine applications. Additionally, it has the ability to use MultiSense to recognize meters of psychological suffering, but rather than guiding intelligent software, this information is presented to a geographically-distant human healthcare professional

Fig. 29: SimSensei virtual health agent (on left) and Telecoach interface concept (on right)

- **The theatre [60]**

Nijholt and Hulstijn have developed an environment that can be considered as a "laboratory" for research on multimodal interactions and multimedia presentation, where multiple users and various agents exist that help the users to find and communicate information. The environment represents a virtual theatre, which allows navigation input through keyboard and mouse events, but there is also a navigation agent who is trying to understand the keyboard natural language input and spoken commands and to give feedback using speech synthesis. There is one more agent, Karen, which allows natural language dialogue with the user.

- **Intelligent tutoring goes to the museum in the big city [29]**

  Coach Mike (Fig. 30) is a virtual staff member at Boston Museum of Science that seeks to help visitors at an interactive exhibit for computer programming, the Robot Park. By tracking visitor's interactions and by using animations, gestures and speech synthesis , Coach Mike provides several forms of support, as orientation plans, discovery support and problem solving guidance, that seek to enhance the experiences of museum visitors.

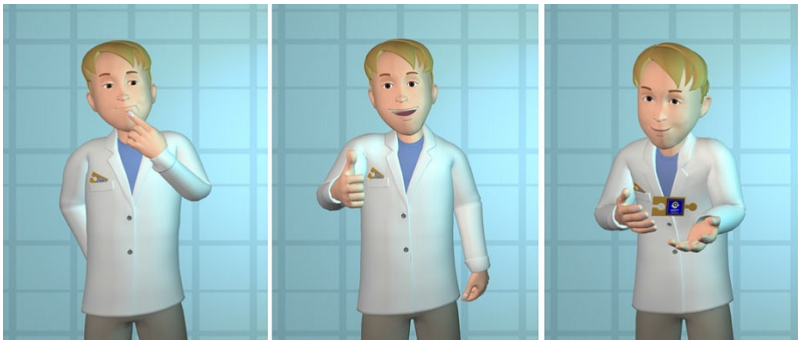


Fig. 30: Mike as 3D cartoon-style pedagogical agent designed to be supportive and understanding

- **SAMIR – web agent [48]**

  SAMIR system is a framework to build intelligent agents for the Web. SAMIR consists of a 3D face which is animated to develop expressions which are recognized by the user, a custom version of the ALICE "chatter -bot" to chat with the user and a learning classifier system to deal with the problem of keeping conversation and face expressions consistent with each other.

- **Assistants for elderly and cognitive impaired users [25]**

  For people with cognitive impairments, who have problems to organize their daily life autonomously, a virtual agent, represented as a daily calendar assistant, can

offer priceless support. Of course this requires that the users of that group will accept such a system and can interact with it successfully. Yaghoubzadeh et al present studies to elucidate these issues for elderly and cognitive impaired users (Fig. 31). The results from this research, through interviews and focus groups, show that acceptance can be increased by way of participatory design method.

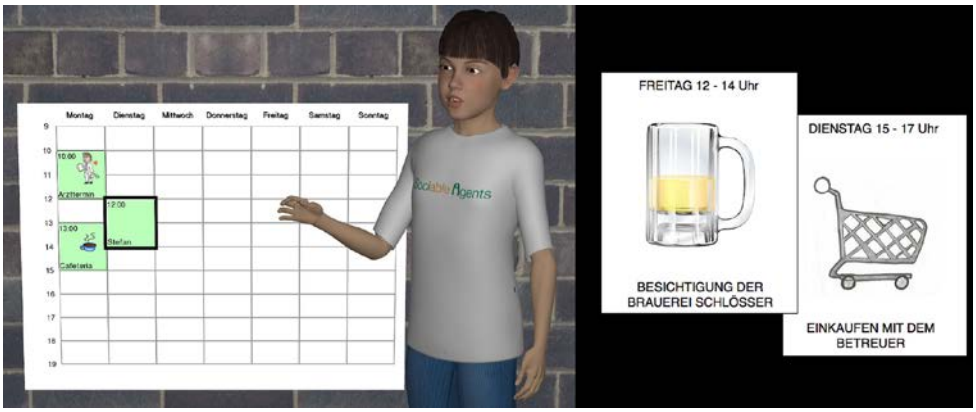


Fig. 31: (Left) Daily assistant "Billie" presenting the user's appointments; Right: appointment. (Right) appointment cue cards used in the study.

3 Motivation and Rationale (Research Objectives)

Various systems already exist that offer diverse features related to virtual humans including conversational abilities, user training, adaptive behavior and virtual human creation, but they constitute limited, "monolithic" attempts that do not provide a comprehensive solution that can cater for broader needs, such as the ones imposed by ambient intelligence environments. In this context, the aim of this thesis is the design, development and evaluation of a generic and adaptable framework that combines all the aforementioned features and employs virtual humans in order to provide interaction, assistance and information provision in ambient intelligence environments. Towards this end, the envisioned framework needs to be highly dynamic and flexible so as to be independent of its context of use. Since the information flow among the three fundamental "components" of AmI environments – i.e., the users, the individual systems and the integrated "smart" environment - is omnidirectional, the system should also be able to communicate with all of them.

There are three distinct roles that virtual humans must be able to support any acting as: (a) assistants for third-party systems; (b) as standalone applications and (c) integral parts of emerging smart environments. When acting as an assistant, the framework should provide the tools needed for presenting information in the form of tutorials and real-time help. Furthermore, the virtual humans should support the users' training on interaction techniques applied for the assisted system. As a standalone application, the framework should provide categorized information visualization and interactive help about the interaction techniques and information presented. Finally, when the virtual humans are embedded in smart environments, the framework should be able to support all the aforementioned functionalities individually or even in combination.

In addition to the above, the virtual humans created through the framework should be expressive and natural in order to be as realistic as possible and enhance the human-oriented character of the system. Therefore, the information provided by the virtual human should be accompanied by body animations and synthesized speech. Additional information may be offered in various forms, including text, images, audio, videos and three dimensional models. These means of information visualization should be dynamic, adaptable and interactive so that different users can manipulate them in different ways.

Finally, a fundamental characteristic of systems embedded in ambient intelligence environments is the adoption of natural user interaction techniques. As interaction with virtual humans in ambient intelligence environments presented in related literature mainly focuses on conversational interaction and user tracking in space, the suggested framework should support but also enhance and extend these techniques by including verbal communication, kinesthetic interaction and mobile devices.

4 Requirements elicitation

This chapter describes the requirements that had to be met for the creation of the system of Bryan as these were modulated through the design process. First, the procedure followed is discussed step by step, stating the different approaches that were adopted in order to achieve optimal results. Subsequently, the requirements are analyzed and grouped into two categories: the functional and the non-functional requirements.

4.1 Elicitation Process

In order for the system to be designed, it was necessary to follow several procedures so as to define Bryan's construction requirements. First, a number of brainstorming sessions was conducted aiming at developing the exact specifications of the system characteristics. These sessions were the first step taken in order to produce ideas as to the scope of a virtual assistant application and also to create a fuzzy overview of the scope of the system. Furthermore, focus groups were scheduled in order to discuss the detailed characteristics that the system should have.

The next step was to assemble the users that were to be interviewed in order to define the basic attributes required for the presentation of the system irrespective of its context. The interviews also intended to define the properties and functionalities that an end user would expect to see in a virtual human system. The people who participated in the interviews varied from 20 to 60 years old, an age group appropriate for the purposes of the project and the decisions that needed to be taken as this is the ideal target group of users of such a system. The results of the interviews were extremely helpful, as a wide variety of features the users expected the system to have was elicited.

Once a rough image of the system was formed, mockups were designed to determine the appearance of the Virtual Assistant. This approach was particularly helpful as it encouraged weaknesses to surface, thus allowing the refinement of the design.

Finally, there was constant evaluation of each new part devised throughout the development of the system.

4.2 User Requirements

Functional requirements are related to the specification of the functions in a system or its component. A function is described as a set of inputs, the behavior and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionalities that define what a system aims to accomplish. Generally, functional requirements are expressed in a form "system must do <requirement>", while non-functional requirements are "system shall be <requirement>".

The functional requirements that were produced by the results of the process of requirement elicitation are the following:

- Multimedia support in order to enhance the user's experience with rich information easily perceived and understood by the user.

- Controllers of information visualization so the user can manipulate the information presented.
- User training of the interaction techniques so as to allow the user to interact with the system and enable them to use the more complex techniques.
- Categorization of information for easier navigation by the users.
- Multilingual information so that any user, no matter their nationality, is able to understand the system's content and, naturally, for the system to attract a wide range of users.

As far as non-functional requirements are concerned, the necessities outlined can be divided into two groups: performance requirements and interface requirements. The interface requirements of the system include:

- The ability to retrieve input from a wide set of sources that describe semantic information
- Interaction should be achieved through the use of natural means of interaction.
- Compatibility with a variety of means of interaction and support of multimodal interaction, using different interaction techniques
    - individually
    - Subsequently in conjunction

The performance requirements refer to the characteristics of the system with a view to the user's achievement of objectives and include the following:

- The system should be self-explaining. Ideally, it should be easy to use for the first time without any special training or at least after three minutes of training
- The interaction techniques should be intuitive and easy to memorize
- The system should be responsive to user actions
- Interaction should be precise
- The overall system setting should be stable and robust
- The system should run on a common personal computer without special hardware requirements

5 System Overview

The framework presented in this thesis supports alternative roles for the virtual humans who may act as assistants for existing systems, standalone "applications" or even as integral parts of emerging smart environments.

Firstly, virtual humans are presented in the role of virtual assistants. A virtual character, Bryan, is presented in order to expose the functionalities the framework offers that can be applied for virtual assistants. Secondly, an overview of some additional implemented scenarios of use is presented, containing virtual humans in different environments and for alternate purposes.

5.1 Virtual Human as an assistant

In the following sections, the fundamental notions of the scene of Virtual Assistant are presented along with their integration into the system. Additionally, the rationale lying behind the decisions taken as to the adopted or rejected approaches is analyzed.

5.1.1 Key Components

5.1.1.1 The Virtual Human

It appears that virtual humans are being increasingly used in helping systems to provide assistance. As defined, virtual assistants are independent contractors which perform a wide variety of tasks. These tasks may be administrative, financial, creative or technical in nature. They are designed to provide customer services, product information, marketing, support, sales, order lacing, reservations or other custom services. Intelligent virtual agents are animated, human-like graphical chat bots. They are embedded with predefined script and responses.

Bryan, is also an animated human-like virtual assistant whose aim is to provide assistance to every application using it as well as to every user handling the application. In the framework applications can modulate the assistance they wish to be given. The assistance obtained from Bryan comes through hand and body movement (gestures) and speech. It can describe any content given in order to provide additional information to the user using text to speech. The application allows the description to be dynamic and adaptive at any point and the description can be enhanced through suitable body movement and gesticulation. Furthermore, Bryan responds to the users' wishes, so when a user gives

specific commands, the system reacts accordingly. By the same token, it obeys to the real time application commands that may come and performs them at once.

5.1.1.2 Structured Tutorial

The term “tutorial” refers to the approach the system has adopted so as to present the user with a guided overview of what they will see if they chose to use the application. Every application can load its own tutorial for the users and change it dynamically as well. The tutorial consists of images and videos with which an application can give more information about specific parts of their system, such as interaction. The application also provides text for speech to the virtual assistant in order to describe each content accordingly. During the tutorial process, the user can interact with the system in real time using the interactive hand cursor. He/she can stop or resume the guided tutorial, can restart the tutorial, can change chapter of interest, can affect the flow of the current chapter, for example, by moving to the next or previous image or pausing the current video, can mute the assistant and just see the information presented and a description below in subtitles, can change the current language so that the information and Bryan’s speech will follow suit.

5.1.1.3 User Interaction Training

In ambient intelligence systems, the interaction techniques have been multiplied by the growth in hardware and the new technological achievements, such as the Kinect sensor. The multimodal interaction has jumped to new paths which, in many cases, are difficult to be understood and used by all, such as the gestures that applications may use, which do not always feel so natural to users so as to recognize their action. Hence, the need of assistance and training generated in ambient intelligence applications. So, Bryan, apart from being a virtual human to assist at anytime, it also provides a training mode that users can utilize to train themselves in the interaction methods used by the application with the aid of Bryan. Bryan executes each interaction technique and then asks the user to act likewise. The application gives a number of opportunities to the user to carry out each technique; if the attempt is successful, then the assistant rewards the user and moves on. If not, it allows extra attempts for accomplishment or proceeds with the next technique. A user can rehearse any of the interaction techniques or go to the previous or next one anytime they wish. This process allows for a number of interaction results which can be used as evaluation findings with regard to the interaction techniques that applications use. These results will be either positive or negative as far the system is concerned, therefore, the developers will acquire valuable information as to which interactions users find difficult and non-usable and which interactions feel natural and straightforward.

5.1.1.4 Categorized Information

In almost every interface nowadays there is a form of filtering information in order for the user to navigate each system more easily and find the information they need everything more quickly. Depending on the application style and needs, the filtering technique can be divided in many ways, by content, type, time etc.

Bryan's system provides a method of filtering information to the applications that use this framework. The filtering is divided in three stages. In the first stage, the filtering is performed at content level where every application defines how many and which categories they will have by giving them a title and their content in images, videos, text, three dimensional models and audio. The application also provides the text to speech for the assistant. In the second stage, the filtering is performed at visualization type level. Each category is divided into types of visualization information such as image, video, text, three dimensional models and audio. Each category can have at least one type of visualization information, with the maximum number being five, which is the total number of all types. In other words, after selecting their preferred category, users choose the form in which they want the information of the category to be presented. The third stage is the information presentation one, in which content from the selected category and in the selected form of visualization is demonstrated. The user can interact with this mode (as with the previous). At this level, they can manipulate the flow of information by going to next /previous item or playing/stopping/pausing/resuming a video or rotating a 3D model. During the information presentation, Bryan gives the description (in speech and gestures) that the application provided it within the content loading stage.

5.1.1.5 Real Time Assistance

Many systems use a way of helping their users to interact with them or to understand their parts. Usually, a reasoning model is used to understand when a user needs help in order to provide it. In Bryan's system, there is an instant technique that allows an application to send commands to the assistant to help the users when needed. These commands vary in complexity and can be overly simple, such as "say this", or too complex as speak X and show Y image/video while you animate Z motion for E time. But although the commands can vary significantly, Bryan should react properly to all of them. Thus, a communication system has been developed in order for the application and the assistant to collaborate, with a set of commands either simple or more complex. Each application has

the control of the assistance and provides it whenever it is deemed (through the rules that have been composed) necessary. This way, Bryan becomes an assisting tool, which is additional and not main to the application process, whose help the application can use whenever such help is needed; otherwise, it stays in the background waiting for commands either from the application or the user.

5.1.1.6 Information Visualization

Every modern system should contain an information scheme along with its visualization for its users. The preferred visualization should be easily understood by the users and have a natural mapping of what it represents, because it is important in human computer interaction that every system should be easy to use and navigation and search with its components to be easily recognizable for what they represent. This will make the system usable, tangible and reliable for the users that handle it. In order for the above to be accomplished, Bryan's system contains components for every type of visualization information. The main types of information are image, video, multilingual text, 3D models and audio. These types are represented in the scene as interactive factors totally identifiable by the users for what they are. Their presentation is displayed on a virtual projection screen. This information is related to content coming from the application and every item in it has its own helping description.

"A picture is worth a thousand words" is an adage often used for **Image.** Through this information type the application can provide genuine representation of its system design. Apart from description of items, photographic memory is common in human brains and has the ability to associate concepts, facts and existing items with visual impressions. Image support is straightforward through the creation of quads rendering sprites in the three dimensional space; this implementation is enhanced in Bryan's system through the use of full screen mode for greater details. In the images list, the user can manipulate the flow with the aid of the previous and next functions as well as the slideshow mode.

Video playback is offerσ the most frequently used operations, play, pause, stop, resume. A user can also navigate through the video list with the previous and next functions. In addition, a full screen mode is provided for further details.

Multilingual text is provided either as main information visualization type or as secondary auxiliary method through subtitles and titles. Except for typical written

multilingual representation, the system offers multilingual voice synthesis on demand using Microsoft Text to Speech technology, which Bryan uses for speech. It is used in Categories or Chapters to give their title as well as the menu's titles, and when in mute mode as subtitles of spoken voice. Additionally, multilingual text provides written letter style information as primary information visualization about the selected category or chapter in tutorial level.

Three Dimensional Models may be manipulated to offer detailed information regarding an item. The handling of the models is accomplished through rotation, offering stereoscopic view of any angle the user is interested in. Maneuver of models is achieved through multimodal interaction and therefore different techniques are supported simultaneously. The user may either virtually "push" appearing virtual rotation arrow buttons to rotate in X and Y axes or verbally give commands for rotation right, left, up and down or even control the rotation of the model through smart phone orientation events.

Audio is provided either through audio effects during the interaction of the user or the interpolation of components and via the voice of the assistant in many languages that can be changed as well. Audio can be turned off and on anytime the user whishes to, so any audio component is off or on accordingly.

5.1.2 Data Modeling

5.1.2.1 Data Storage

The initial idea of data modeling was to save the data needed in a configuration file, dynamically store the data in structures and retrieve all the information from them. This was later considered a bad decision because each time additional information was required, the effort of adding them to the system was enormous and generally it was not an efficient way of data manipulation from code and composition perspective.

In computer science, in the context of data storage, serialization (81) Is the process of translating data structures or object state into a format that can be stored (for example, in a file or memory buffer, or transmitted across a network connection link) and reconstructed later in the same or another computer environment. When the resulting series of bits is reread according to the serialization format, it can be used to create a semantically identical clone of the original object. This process of serializing an object is also called marshalling an object. Bryan's system does not impose any limitations regarding the context of use. Therefore, the need for support of any type of information arises as long as it

is in a specific formal format. Therefore, the system uses xml serialization for data storage and retrieval.

Entities are organized in classes inside the project. Each of the basic components (assistance, information visualization) is represented in its own class designed to fit the needs of information data management.

A Tutorial contains a list of:

- A name for the multimedia elements path
- A dictionary of language and sequence of strings that represent the speech for each language
- An animation for Bryan's motion
- A multimedia id , between image,video,3D model, text, audio
- A time duration in seconds
- A chapter title

A Training contains a list of:

- A dictionary with key: animation that represents the interaction technique and value: a dictionary with language and the description of each interaction
- A number of attempts per interaction technique
- A dictionary of animation and their names for chaptering

A Category contains a list of:

- A dictionary for category name per language
- A list of scripts that will be used in this category
- A dictionary for image speech per language
- A dictionary for video speech per language
- A dictionary for 3d model speech per language
- A dictionary for text speech per language
- A dictionary for audio speech per language
- A dictionary for path name per information visualization type
- A name of the representative item path ; this item can be a 3D model or an image

A Grid contains:

- The grid's name

- A number of grid items, referring to the categories that it has
- A name of the preferred shape representation of the grid, between circle, horizontal lane, vertical plane and carousel

A Language grid contains:

- A dictionary of flag image per language
- The available languages
- The folder path for the flags retrieval

A Harvester contains:

- The gestures that the smart phone will use for the interaction with the system

For many complex objects, such as those that make extensive use of references, this process is not straightforward. So in order for date to be manipulated with more multifarious entities, one first prototype of ontology based system was created for the data manipulation of Bryan's system and is planned to be used in the near future.

5.1.2.2 Data Retrieval

Data stored in the Serializable Xml can be retrieved via the process of deserialization which is the opposite operation of serialization, extracting a data structure from a series of bytes (which is also called unmarshalling). Through deserialization all the stored data are loaded in structures and in real time procedure anything that is needed for the information presentation is gained from those. Furthermore, the system allows the addition of information at runtime. Apart from trivial changes such as one more multimedia element, for instance, an image or a video, Bryan's system allows the importation of complex components such as an entire category or a chapter in tutorial. At the time that new information is added to the system, the required actions are taken to recalculate all required information and adapt the visualization to fit the updated information. Finally, the ability to change information of elements is offered in the case of mistakes or updates at runtime.

5.1.3 System Design

Bryan's system was created using an iterative design process during which constant evaluation was conducted. Although the system's general concepts remained constant, various dilemmas arose due either to limitation constraints or to more efficient techniques discovered along the implementation. Constant evaluation with both expert and novice

users in a wide range of ages was carried out in order to arrive at the most suitable approach for each design issue.

In the following sections the design of the system is presented not only by stating the final results, but also analyzing the advantages and disadvantages of the different routes examined and stating the rationale of the decisions taken regarding adopted and rejected ideas.

5.1.3.1 Visualization of Basic Design Components

Bryan's system is an assisting tool which can be additive to existing systems or perform as a standalone application for information and assistance. Because of the fact that the system's basic purpose of use is to offer support, then, its design should be as easily understood and as manageable as possible. Therefore, the initial design decision was focused on the motto "Keep it simple". Simplicity does not mean that the system is poor in graphical design, but on the contrary, it is rich in scene components and 3D visualization and animations. Minimalism lies in the way all components are displayed to the user while interacting in order not to perplex his/her thoughts of what he/she is going to face.

To comply with the design decisions, the basic graphical components selected are the Virtual Assistant Bryan, the scene as a room representation to give a friendlier environment, the projection screen for viewing the information content, some common buttons such as mute/un-mute, return to previous level and language menu for changing language at runtime, and the hand-cursor, an interactive tool for the user to select items of the system. Obviously, the camera could not be missing as it displays all the above to the users' eyes as well as the lights which give a more pragmatic view of the entire scene.

The most basic component of the entire system is the **virtual human**- **Bryan** (Fig. 32), because it is present in every stage and communicates with the application as well as with the user at every level and in different ways. As it is the main tool for the user to interact with the system, the plan was to make it look human as much as possible in order for the users to feel more comfortable. So, Bryan is a virtual human with humanoid behavior, that speaks and moves as common people do. Oral feedback is provided through the animation of his face and mouth. The voice is quite natural and representative of the attitude and the general behavior the system wants to promote, that is sociable, informative and helpful. Its appearance is neither too formal nor too casual. Its gender is male but it could also become female just by changing the model and the voice. In addition, if the application is used by children, it could take the form of a young child as well. The

animations of the body reflect the movements that the virtual human performs to communicate the assistance, such as welcoming the user or saying goodbye, showing information and items, and looking to a specific area.

Fig. 32: Bryan, the virtual Human (left), T-Pose of the Virtual Human (right)

The **projection screen** (Fig. 33) is another important design component because a large proportion of the information visualization is displayed on it, for instance, images, videos and text. As the entire system needs to give the impression of an informative, helpful and training environment, the information is displayed on a projection screen as a typical presentation. So, in a sense, the assistant is like a “teacher” who presents the “lesson” onto a projection screen. As in a typical presentation, there are various ways to browse through it. For example, if the presentation is an image slideshow, it can be paused or sent forward and backward whenever the user wishes to do so. These ways are part of the information visualization components controls and will be analyzed later.

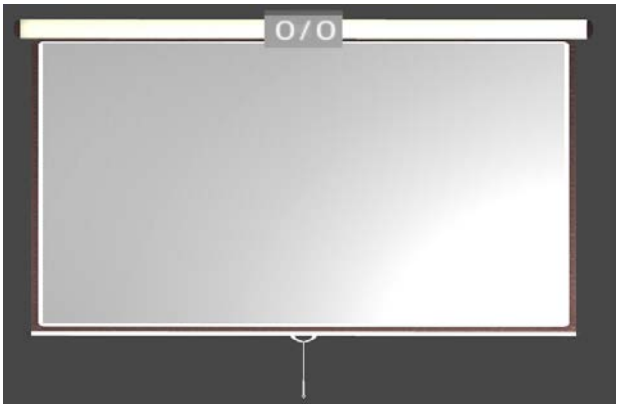


Fig. 33: The Projection Screen

The **environment** (Fig. 34, Fig. 35) that encloses all the design components is represented through an ordinary room with walls and pictures on them so that the users feel as if they were at home, or, at least, feel that they communicate in a familiar environment. A **camera** is the component that brings the entire scene view to the users' eyes. It is a dynamic camera with animation techniques in position and field of view. The user explores the world through the camera and enters every state appropriately and at the right time. Whenever needed, the camera hides objects from the users view. **Lights** surround the scene to give a more realistic graphical environment with shadows on the objects. The dynamic lights are interwoven with the camera's position, therefore they move accordingly. Naturally, static lights, which illuminate specific areas of the scene, are also included.

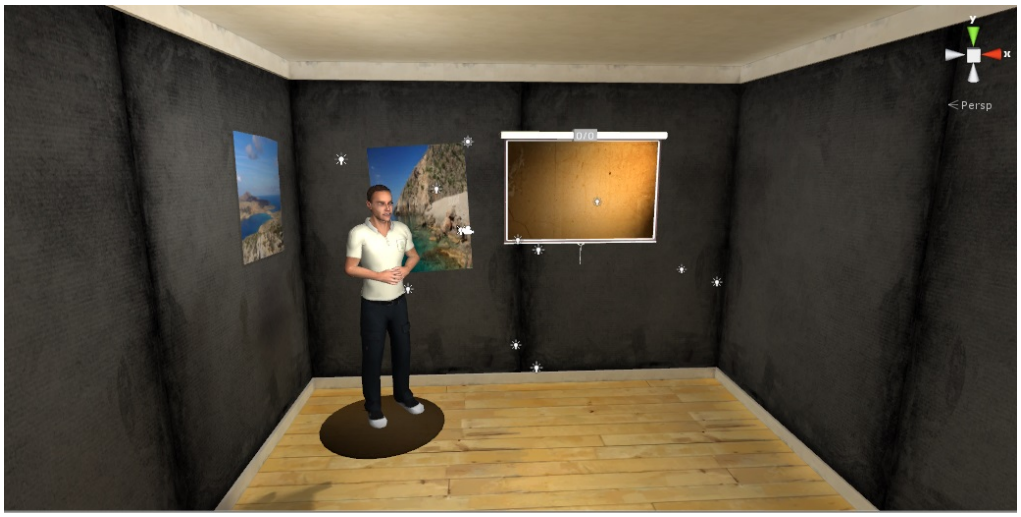


Fig. 34 : A view of Bryan's room

Fig. 35 : Another aspect of the environment

A number of general scene components that are present in every stage of the project there are some interactive buttons that contain animations on their idle, hover and click states and which provide actions that affect the entire environment such as the **mute button** that disables all the sounds in the environment as well as the assistant voice and the same time **subtitles** which are displayed in a field of the scene. There is the **un-mute button** that restores the sound. The **return button** provides the functionality to browse through the several states. As a button, it includes animations on hover and click states to give visual feedback to the user who interacts with it. (Fig. 40,Fig. 41,Fig. 42)

One more interactive and basic to the content component is the **language selection** through which a user can change the current language any time or stage and as a result of this action, change all the interdependent components, such as the titles and text as well as the assistant's speech and voice. When the language button is selected, then a menu of all the available languages is displayed through an interactive flag grid for the user to choose. The grid contains flag buttons which are interactive and provide animations on hover and click. The hover action is represented through a soft undulation and the click makes the flag strain and become a parallelogram (Fig. 37, Fig. 38, Fig. 39). At the beginning of the interaction of the system, in the idle state, the same language grid menu is presented to the user to select the preferred language. (Fig. 36)

Fig. 36 : Language Menu

Fig. 37 : Flag's idle state

Fig. 38 : Flag's hover state

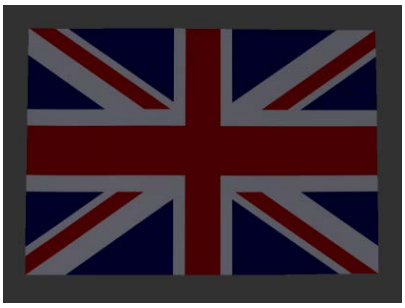


Fig. 39 : Flag's selection state

Fig. 40: mute sound

Fig. 41 : un-mute sound

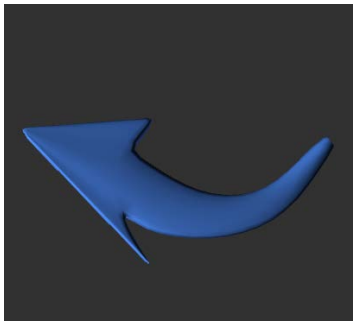


Fig. 42 : return to previous level

Fig. 43 : previous item

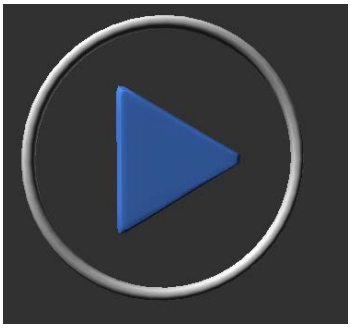


Fig. 44 : play video

Fig. 45 : next item

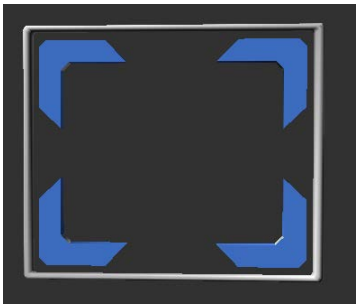


Fig. 46 : enter full screen mode

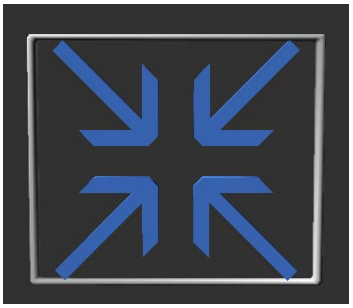


Fig. 47 : exit full screen mode

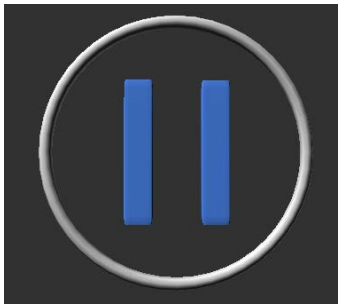


Fig. 48 : pause video

The user employs a **virtual hand cursor** (Fig. 49) to communicate with the system and select items. This hand cursor moves along the user's hand and follows precisely the movements of the user's hand. The goal is to give the user the impression that they actually position their real hand on the scene and that they are in control of the interaction in every way. The hand cursor contains a clock that reports the time that a user is over an object. If the time of the clock "fills", then the object is "clicked"- selected and the hand grasps it. The hand cursor has several states, idle, hovers and clicks according to its location in the scene. The clock on the hand is dynamic and can be removed.

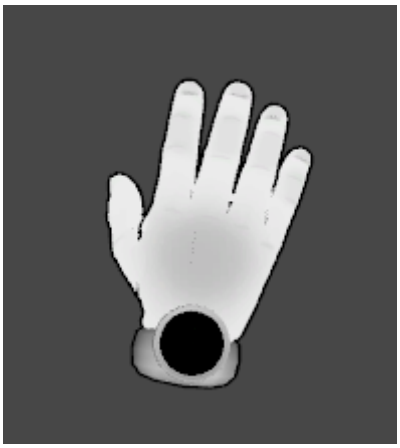


Fig. 49 : The virtual hand cursor

5.1.3.2 Assistance Visualization

Apart from the general scene components that were presented above and the language selection, another menu is displayed that contains three basic sub-menus that enclose and represent the assistance and information the system provides. Those three subsystems are tutorial, training and information categories. The user can choose any one of these three categories and look at the information provided (or return to the previous level through the return button or leave the area). The design decisions of the three submenus are the following subchapters.

5.1.3.2.1 *Tutorial Visualization*

This section of the system describes a structured tutorial which provides information given by the application that uses the system in order to give more instructions for what the application represents, does and how one interacts with it. The whole presentation of the tutorial is divided into three areas. First, there is the assistant's area where the assistant stands on the left side of the scene and through body and hand animations as well as face motions it presents the information, providing realistic communication with the use of speech in the preferred language and through a believable voice. Secondly, the central region is characterized by the display of the information on a projection screen. This gives the feeling of a real presentation with the human on the one side and the projection on the other. On the projection screen, the number of current information items is displayed, for example 3/7 is the current one.

The information is displayed through images, videos and written text. For every type of information, a controller is provided to manipulate the flow of the information. There are three controllers, the image controller, the video controller and the text controller. The **image controller** has three functionalities: go to next image or to the previous one, enlarge

the image, which is practically to zoom in to the image for more details and zoom out if it is in zoomed state, and restart the slideshow. The **video controller,** apart from the image controller functionality, contains play and pause for controlling the video run. The **text controller** (Fig. 43,Fig. 44,Fig. 45,Fig. 46,Fig. 47,Fig. 48)is the same with the image controller and, it provides, as an additional functionality, an acapella-like rendering of the text while the assistant reads it. As a result, people with hearing issues will be able to understand the virtual human speaking. Of course this functionality can be enabled or disabled anytime while the text is rendered on the projection screen.

These functionalities can be achieved with the representative buttons that are displayed below the projection screen, which have animations on hover and click mode as all interactive buttons, or with gestures from the user's hands, or even through mobile gestures.

The third part of the tutorial is the chaptering. With this method, a catalog of chapters is displayed on the one side (right side) of the scene containing all the tutorial information divided into chapters. Each chapter has its title in the selected language and a representative background image. Thus, if the user does not wish to hear the entire tutorial presentation, they can just select the chapter of their interest and the information of this chapter will be displayed automatically on the projection screen and the assistant will start the presentation. The chapters are created by the application that uses the system and the system assumes to present them as a list to the user. If the list is too long so to fit the scene height, then the display becomes scrollable and the user can choose to go up or down the list with the aid of two representative interactive arrow buttons. The mute button is always on the scene so that the user can mute the voice of the assistant, in which case, subtitles of the spoken text are viewed automatically on the bottom of the scene, as well as turn off the sound of the video. There is also an un-mute button which can revert the mute action, restore the sound and hide the subtitles. The return button is in its usual position and can take the user to the previous level – into the three basic sub-menus choice. (Fig. 50)

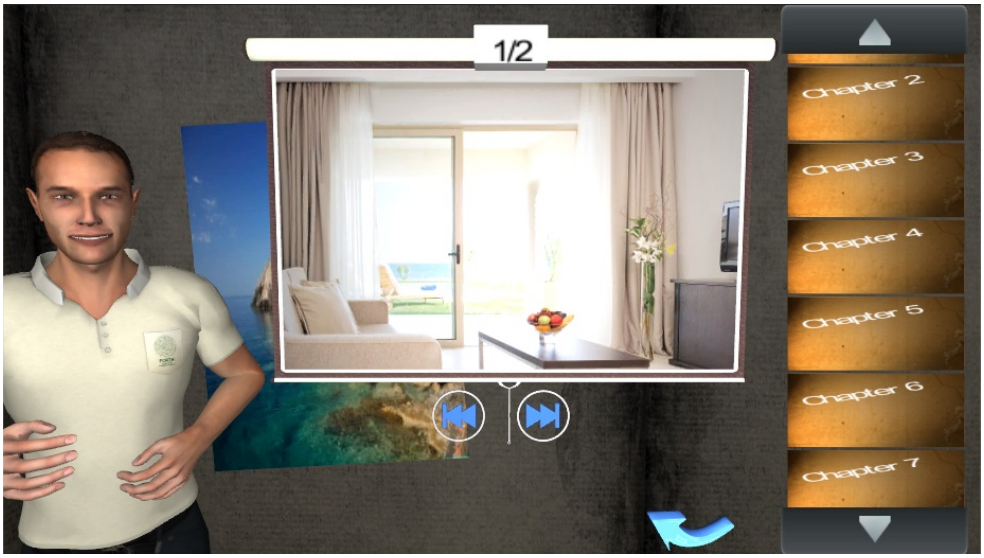


Fig. 50 : The Tutorial View

5.1.3.2.2 *Training Visualization*

The training system option is a method with which the assistant teaches the user the interaction techniques of the application in order to help them learn some difficult practices the application uses in interaction and which are not so easily understood by the user or practices frequently used as interaction techniques in ambient intelligence environments. Because of the fact that this procedure is used as a training tool, the first goal was to make it as simple as possible and not to complicate the user with complex scenes. Therefore, the scene is divided into two areas: the assistant area and the interaction techniques list area.

In the assistant area, the virtual human is presented in full body view in order to perform any interaction technique needed either with hand and full body gestures or with speech commands (Fig. 51). The process is as follows: the assistant begins to explain to the user that 'he' will be the presenter and teacher of the interaction techniques of the application. Subsequently, the first technique is presented with the assistant demonstrating the gesture animation needed for the interaction described to occur, and explaining, at the same time, the purpose of this gesture and where it is used in the application. Then the assistant addresses the user asking them to execute the same gesture which will enable him/her to interact with the system. After that, the user has a number of opportunities (the number is defined by the application) to succeed in the interaction technique. If the user succeeds in his/her attempt, then it rewards them verbally and continues with the next technique. If not, the assistant speaks complimentary words for as long the user attempts this technique. However, if the user fails in all his/her attempts, then the assistant proceeds with the next method showing disappointment.

Fig. 51 : Bryan in training view performing a gesture

In the training technique list area, a catalog of the available methods that the assistant teaches can be found. Each method has its name-title. Users can select a technique anytime they wish to practice and the assistant gives the provided instructions. The list of the provided techniques along with their spoken description is created by the application. If the list is too long to fit the scene height, then the display becomes scrollable and the user can choose to go up or down the list with the aid of two representative interactive arrow buttons. The mute button can turn off the voice of the virtual human, at which point subtitles are displayed at the bottom of the scene (where subtitles usually are displayed in films). The un-mute button restores the sound to its previous state and hides the subtitles. The return button takes the view to the previous level which is the three basic sub-menus choice.

5.1.3.2.3 *Categories Visualization*

This subsystem represents the categorization of information of any application that uses the system. Each application creates the categories into which the information will. Each category displays an information section. In the “Categories” selection, a horizontal grid comes forward with the categories in a row in front of the virtual human at the bottom of the scene. Each category has its own title and its representative item, which can be a three dimensional model or an image displayed in a three dimensional frame model. This model is given by the application in the data storage level. The view of the grid is also defined by the application; the other types of grid representation are cylindrical, vertical and carousel style. Each category component is dynamic, interactive and has its own animation states (idle, hover, click). In the hover category, the representative item is shown and in the category selection a semicircular menu surrounds the category item with the presentation types of the information. (Fig. 52) Those types can be image, video, text, three dimensional object and audio. A category can have at least one type of information with the maximum number set at five. For the representation of these types three dimensional animated models are displayed. Each one of them represents the type of information that it refers to: a photograph for images, a video clap for videos, a book for texts, a 3D axis system with a 3D object on it for three-dimensional models and a music note for audios. All those models were chosen because they are easily understood for what they represent by the users. They all have button style behavior, although they are 3D objects. They have animations on idle, hover and click state as a feedback to the user interaction. When another category is selected, the menu of the previous selected one is hidden and the current’s category menu is presented.

Fig. 52 : Selecting to view images in a category

Upon selection of image (Fig. 53), video or text type of information, the category menu hides and the camera moves to the scene where the projection screen is on the one side (right side) of the scene and the virtual human on the other (left side). During the camera movement, soft music has been included to make the transition more enjoyable. Then, the projection screen opens and starts to display the information in the type selected. The controller of the information takes its place below the projection screen. It is the same controller for every type as in the tutorial sub-system. The mute, un-mute and return buttons are also in their usual positions as well as the language change button. The virtual human in every image, video or text describes both verbally and with body animations what the item displayed shows.

Fig. 53 : In Image Selection

Upon selection of the three dimensional model, the category menu is hides and the camera moves to the scene where a three dimensional model is on the one side (right side) and the virtual human on the other (left side). Throughout the camera transition a gentle music creates a more pleasant view alternation. The model surrounds three dimensional animated rotation arrows with which the user can manipulate the model's orientation and see more details. It has also the same controller as the image one to navigate to the next or previous model and to enlarge the view of each model for more details. The assistant provides information through speech and body animations for each model.

Upon selection of audio, the category menu is hides and the audio starts to sound. Depending on the sound, the virtual human may speak or not. There is an audio controller

for navigation to the next or previous item and play - pause functionality. A volume control with buttons for increasing or decreasing the volume is also provided. All the above controls are button-styled and have animations on idle, hover and click modes.

5.1.4 Task Analysis

The following sections describe the tasks that can be accomplished using Bryan's system. The Hierarchical Task Analysis (HTA) of the available tasks and their sequence is grouped according to the view of the assistance. The execution of each task is not necessarily sequential, as, in general, users have the ability to combine different functionalities in the order they wish.

Due to the large extent of the diagram, some tasks are further analyzed in separate diagrams in order for them to be clear. These tasks are underlined in the overviews of the hierarchical task analysis.

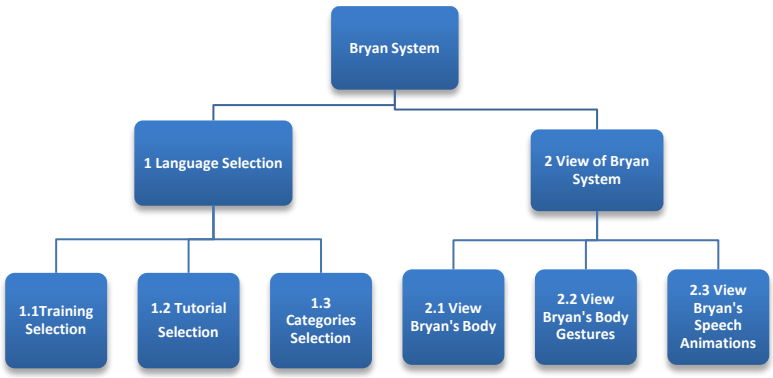


Diagram 1 : The provided Assistance Hierarchical Task Analysis (HTA) of Bryan's system

Assistance Diagram Plan

Plan 0: Execute 1 or 2

Plan 1: Execute any of 1.1 to 1.3

Plan 2: Execute any of 2.1 to 2.3

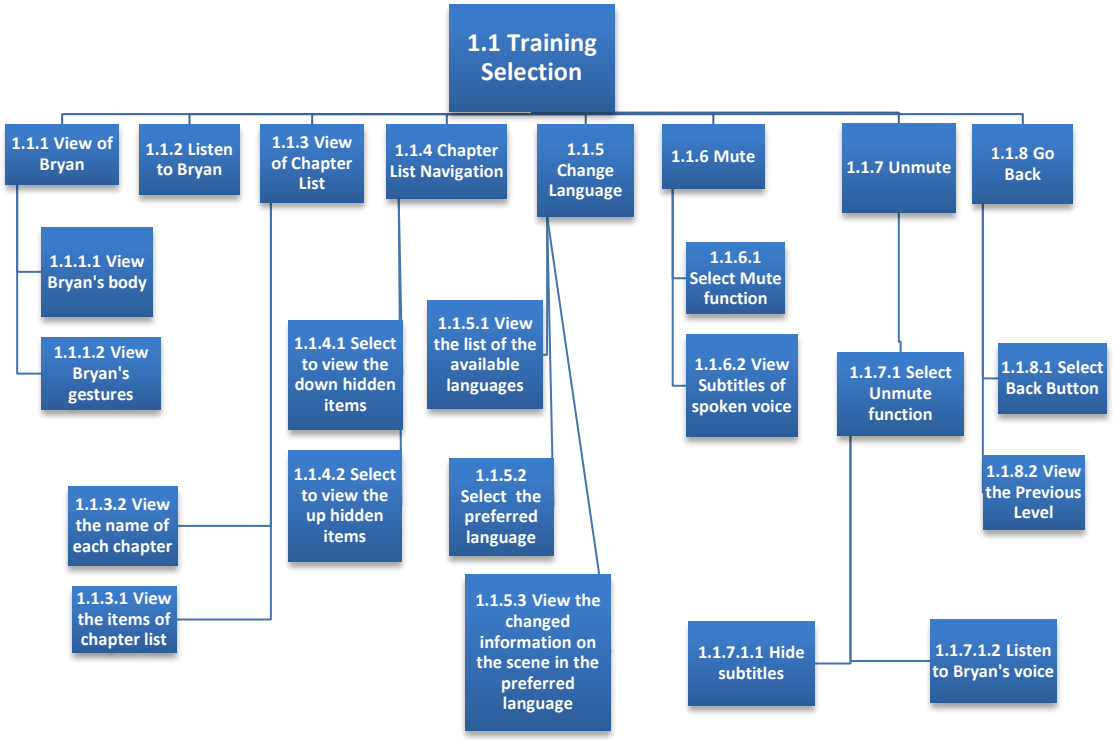


Diagram 2: Training Process HTA

Training Diagram Plan

Plan 1.1: Execute (Execute 1.1.1 – 1.1.8 in any order)

Plan 1.1.1: Execute 1.1.1.1 or 1.1.1.2

Plan 1.1.3: Execute 1.1.3.1 and 1.1.3.2

Plan 1.1.4: Execute 1.1.4.1 or 1.1.4.2

Plan 1.1.5: Execute 1.1.5.1 to 1.1.5.3 sequentially

Plan 1.1.6: Execute 1.1.6.1 and 1.1.6.2

Plan 1.1.7: Execute 1.1.7.1, 1.1.7.1.1 and 1.1.7.1.2

Plan 1.1.8: Execute 1.1.8.1 and 1.1.8.2

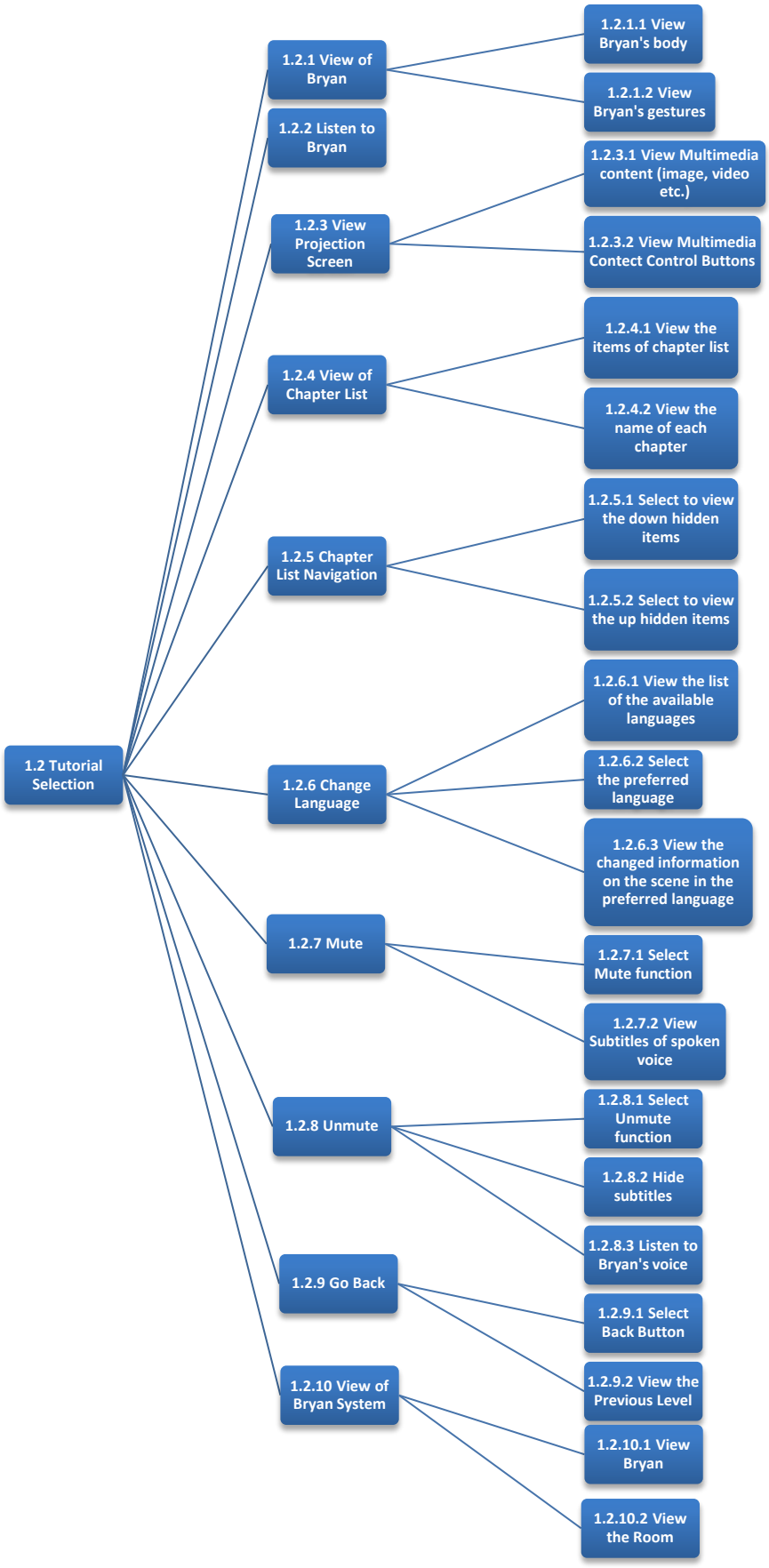


Diagram 3: Tutorial Process HTA

Tutorial Process Diagram Plan

Plan 1.2: Execute any of 1.2.1 to 1.2.10

Plan 1.2.1: Execute 1.2.1.1 and 1.2.1.2

Plan 1.2.3: Execute 1.2.3.1 and 1.2.3.2

Plan 1.2.4: Execute 1.2.4.1 and 1.2.4.2

Plan 1.2.5: Execute 1.2.5.1 and 1.2.5.2

Plan 1.2.6: Execute 1.2.6.1 to 1.2.6.3

Plan 1.2.7: Execute 1.2.7.1 and 1.2.7.2

Plan 1.2.8: Execute 1.2.8.1 to 1.2.8.3

Plan 1.2.9: Execute 1.2.9.1 and 1.2.9.2

Plan 1.2.10: Execute 1.2.10.1 and 1.2.10.2

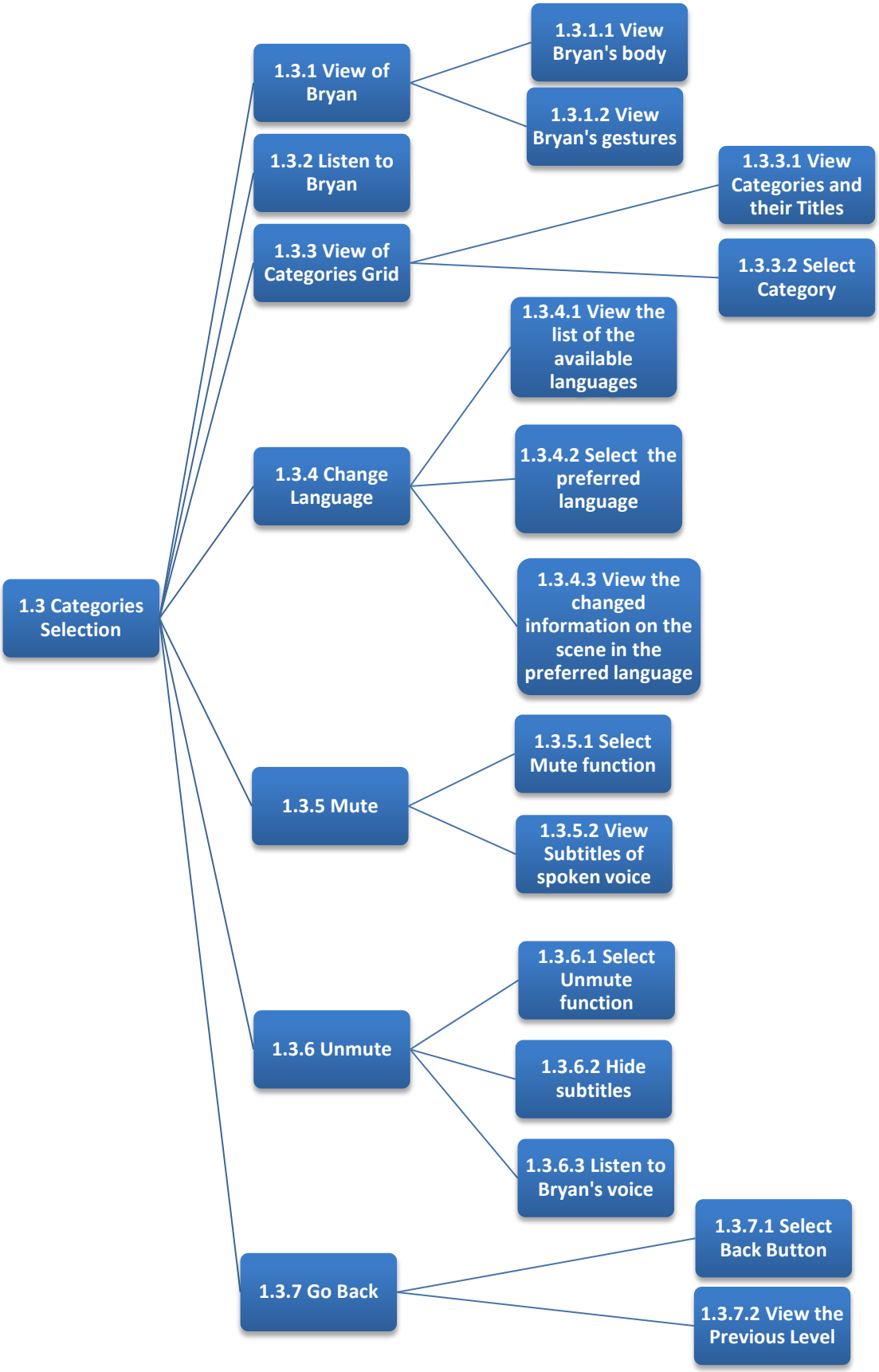


Diagram 4 : Categories process HTA

Categories Diagram Plan

Plan 1.3: Execute any of 1.3.1 to 1.3.7

Plan 1.3.1: Execute 1.3.1.1 and 1.3.1.2

Plan 1.3.3: Execute 1.3.3.1 and 1.3.3.2

Plan 1.3.4: Execute 1.3.4.1 to 1.3.4.3

Plan 1.3.5: Execute 1.3.5.1 and 1.3.5.2

Plan 1.3.6: Execute 1.3.6.1 to 1.3.6.3

Plan 1.3.7: Execute 1.3.7.1 and 1.3.7.2

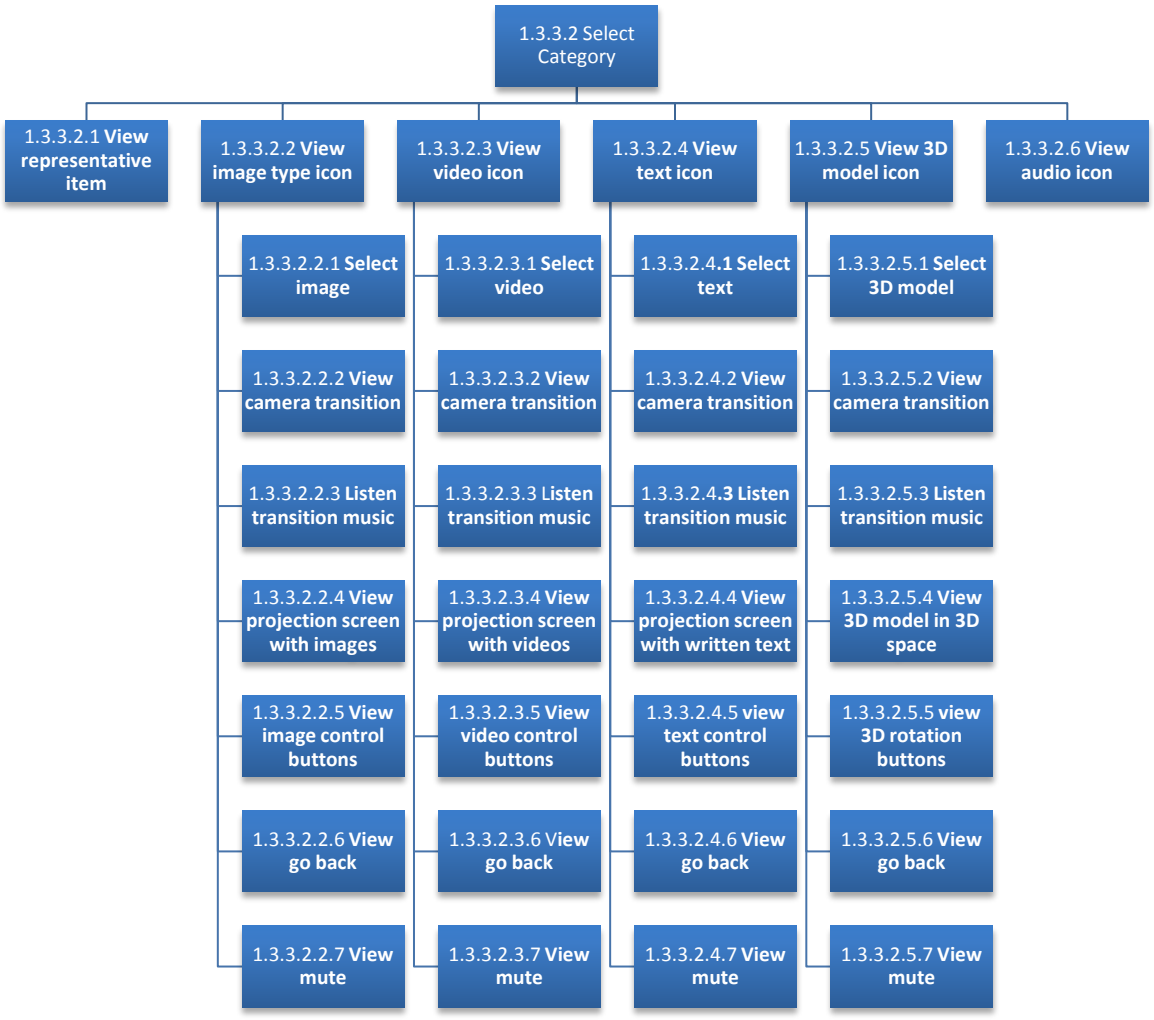


Diagram 5 : Multimedia Type Selection HTA

Multimedia Type Selection Diagram Plan

Plan 1.3.3.2: Execute any of 1.3.3.2.1 to 1.3.3.2.6

Plan 1.3.3.2.2: Execute 1.3.3.2.2.1 to 1.3.3.2.2.4, then 1.3.3.2.2.5 to 1.3.3.2.2.7

Plan 1.3.3.2.3: Execute 1.3.3.2.3.1 to 1.3.3.2.3.4, then 1.3.3.2.3.5 to 1.3.3.2.3.7

Plan 1.3.3.2.4: Execute 1.3.3.2.4.1 to 1.3.3.2.4.4, then 1.3.3.2.3.5 to 1.3.3.4.3.7

Plan 1.3.3.2.5: Execute 1.3.3.2.5.1 to 1.3.3.2.5.4, then 1.3.3.2.5.5 to 1.3.3.5.3.7

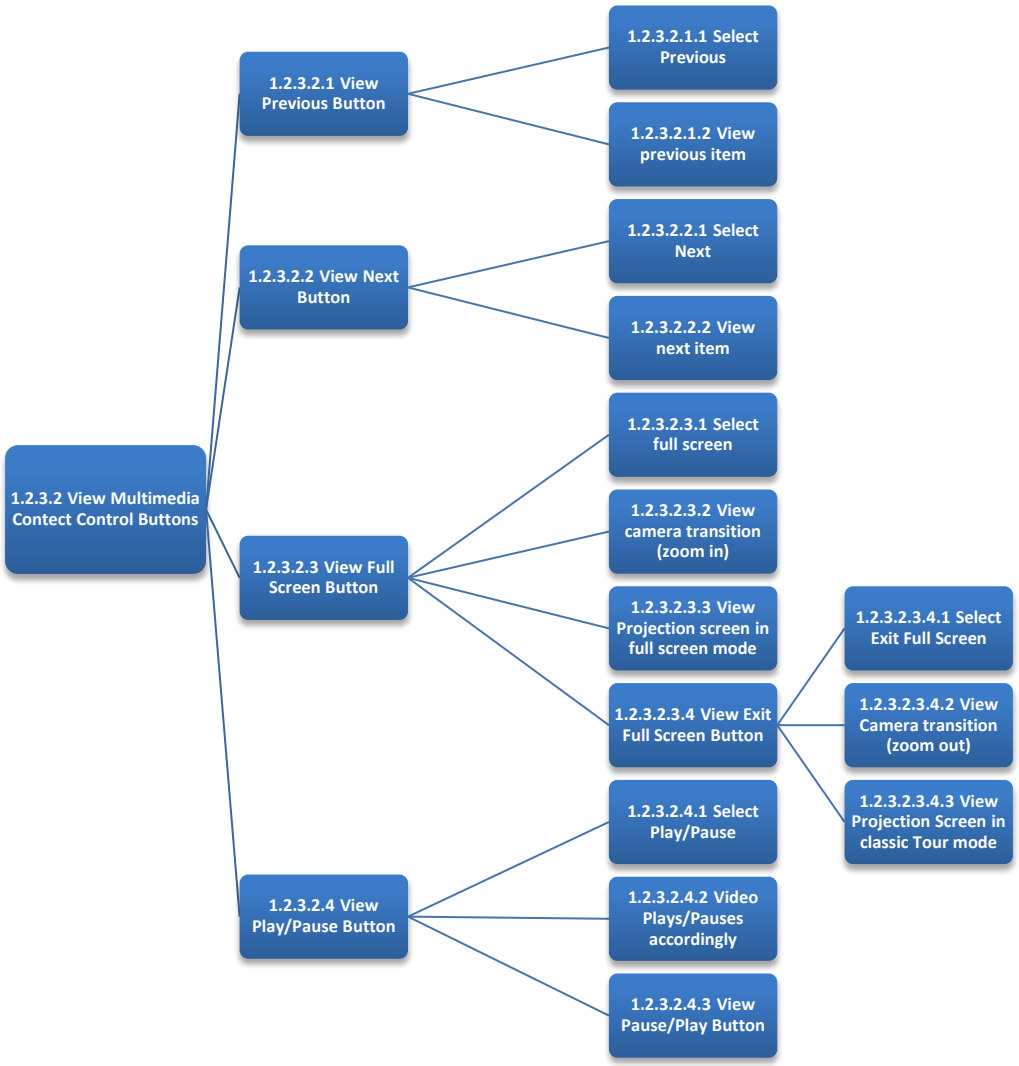


Diagram 6 : Multimedia Content Controls HTA

Multimedia Content Diagram Plan

Plan 1.2.3.2: Execute any of 1.2.3.2.1 to 1.2.3.2.4 (MOVE DIAGRAM4 after tutorial process)

Plan 1.2.3.2.1: Execute 1.2.3.2.1.1 and 1.2.3.2.1.2

Plan 1.2.3.2.2: Execute 1.2.3.2.1.2 and 1.2.3.2.2.2

Plan 1.2.3.2.3: Execute 1.2.3.2.1.3 to 1.2.3.2.3.2

Plan 1.2.3.2.3.4: Execute 1.2.3.2.3.4.1 to 1.2.3.2.3.4.3

Plan 1.2.3.2.4: Execute 1.2.3.2.1.4 to 1.2.3.2.4.3

5.2 Case Studies

In the following case studies virtual humans are presented that provide real time assistance to users, act as standalone systems and are able to communicate with other agents in large scale environments.

- **Smart Hotel Room Virtual Butler**

In the context of creating a smart hotel room [71] a three dimensional virtual assistant (Fig. 54) was created in order to provide assistance regarding the room's smart functionalities which the user is not aware of. The virtual assistant is displayed in a television located in the room's lobby.

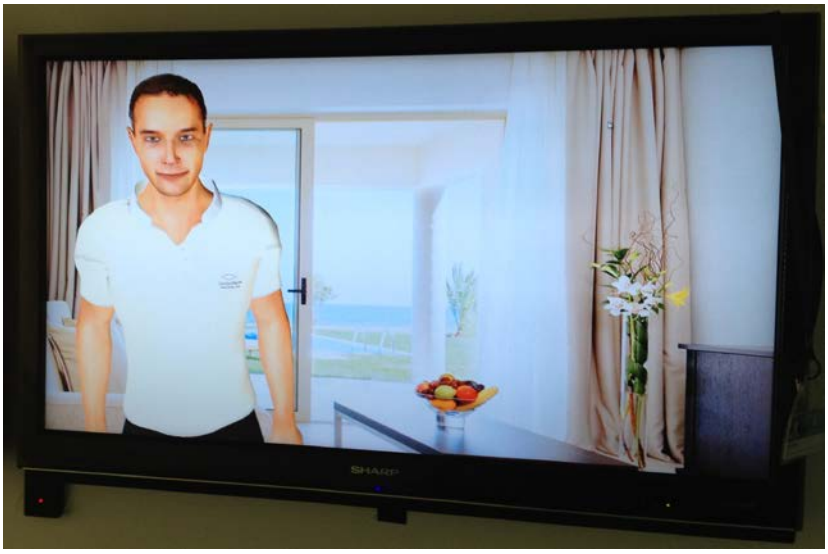


Fig. 54 : Hotel Room Virtual Assistant

The assistant is a virtual human able to speak in various languages since the room hosts visitors of different nationalities. The assistant's multilingual speech is enhanced with body movement and facial animations. Moreover, the virtual human is capable of making full-body gestures in order to point to specific areas.

The assistant communicates with the environment and retrieves information when needed. For instance, the assistant is informed of the users' personal details (name, age. Gender, etc.) and language spoken at their first visit in order to guide them accordingly.

Room guidance is performed during the visitors' first entry but may be performed again at any other time upon request. During the guidance the assistant provides guests with information regarding the room's points of interest (Fig. 55) through the combination of speech, gestures, images and videos. Moreover, the assistant presents the provided means of interaction so as to manipulate the system. For instance, the user is able to open or close window's blinds by pointing with the remote control at them and to increase or decrease lightning by raising or lowering his/her hand. Furthermore, the user is informed about the room's automatic actions (e.g. the lights turn on automatically when the guest gets out of bed at night) along with the available manual settings (e.g. the users can disable the automatic lightning at night) using a portable device. In general, the guidance aims to inform the guest of the intelligent room's capabilities and how to handle and control them on demand.

Fig. 55 : Hotel Room Smart Points of Interest

Additionally, the assistant informs the guests that they are capable of leaving messages to the personnel which are presented by the assistant. Finally, the users are informed that they may place their laundry on a specified location and the personnel are automatically notified for their ingathering.

As a whole, the assistant is able to communicate with the environment through a room reasoner which is able to perceive the users' actions, make the necessary decisions and distribute the needed instructions to all the system's components, including the virtual assistant.

The inputs the user can have are the following:

1. Be informed about the guest's arrival and further personal details
2. Guide users at their room during the first time and on demand
3. Inform the guests of emergency situations
4. Muting the sound and if so, display subtitles
5. Inform the cleaning personnel about potential messages from the guests
6. Inform the cleaning personnel about the laundry, if any
7. In case of forgetting the laundry inform the cleaning personnel
8. Update guests with the status of the laundry
9. Notify personnel about the status of cleaning requests (initiated or canceled)
10. Inform guests about the weather, hotel offers and other information users might be interested in
11. Notification that the guests choose not to be disturbed

The assistant was implemented using OpenSceneGraph [77] and C++ programming language and was the first step towards the development of this thesis.

- **Door Welcome and Identification Agent**

This is a system that welcomes users in Ambient Intelligence Facility and asks to identify in order to provide them with access to the building via voice recognition and speaker identity verification.

Upon approaching the entrance door, the virtual three dimensional agent (Fig. 56) welcomes the user through speech and hand gestures and asks him/her to tell the password in order to identify his/her valid access to the building. (Fig. 57) After the provision of the user's password, speaker identity verification is responsible for accepting or rejecting access to enter the building, taking into consideration both the identity of the speaking user and the accompanying password. Bryan opens the entrance door only if the user is registered and the password matches the corresponding user's password. In the case of failure to identify successfully, the system allows the user to try again up until successful identification.

Fig. 56 : The Door Agent (left). The agent welcomes the visitor (right)

Fig. 57 : Visitors in interaction with the agent (left). The agent records the visitor's password to verify the access. (right)

- **Agent to Agent Communication**

In this scenario of use information and interaction can be personalized through agent to agent communication. This scenario involves the communication between three agents to facilitate assistance in a building's different rooms. The concept involves an agent protocol communication as well as employing users to pass messages on different rooms, so as to motivate them to explore various interactive exhibits. The first agent is positioned at the entrance and acts as a "doorman" allowing entrance to certified users only. The second agent is located at a room containing interactive games and the third one is in the lobby of the room where TimeViewer is employed.

The first agent recognizes the users' movement, welcomes them and asks for their entrance verbal password in order to allow them enter the building. If the user's voice and password are recognized and matched to registered entries, the agent opens the door offers personalized recommendations for interactive exhibits that the user may be interested in.

The second agent is in a room full of interactive games, including a jigsaw-puzzle, a word search puzzle, an arkanoid-style game and Paximadaki [63], a game which employs a user's virtual shadow in order to collect rusks.

The third agent is used as an assistive system to TimeViewer, providing overview of the system's design and information on how to manipulate the system.

In the case where a child is recognized, the agent-doorman suggests going to the gaming room and telling the agent "I am a child". Moreover, the agent provides instructions on how to find the gaming room. The child, excited by the prospect of playing games, goes directly to the gaming room and start playing the jigsaw puzzle. The child plays all the available games but spends most of the time playing Paximadaki. Therefore, the agent perceives that the child is excited with the specific game and suggests going to TimeViewer and telling the agent "I want to play". As a result, when told the specific phrase the third agent understands that interacts with a child. Therefore, since children enjoy kinesthetic interaction, the agent trains the child to manipulate TimeViewer using gestures and urges it to interact with TimeViewer. Additionally, the agent instructs TimeViewer to switch to the "tunnel view", as it is more immersive and fun to interact with in comparison to the two dimensional display.

The messages passed between agents are meant to broadcast the users' preferences among different systems in order to provide personalized information according to each user's needs. These messages may either be sent directly from one agent to another or involve the user as a playful task through verbal or non-verbal communication.

6 Multimodal Interaction

A diversity of interaction techniques were examined in order to conclude as to the most appropriate one for the manipulation of the system in a natural way. A major aim of the system was the creation of an affordable system that may be set up using an ordinary

budget which is context-aware and offers means of interaction that are applied in everyday human communication.

Because of the fact that the system contains humanoid characters and the main component of the whole project is the virtual human, the use of voice interaction is a means to achieve a more intimate communication with the system. The ideal vocal interaction would be the real time conversation, in which the users speak and the virtual humans understand and respond to the users appropriately through speech. Here the voice is used as a secondary means of interaction because the system for speech recognition is still in an experimental stage. But the goal is the creation of a fully conversational system and user-virtual human communication through real time speech recognition and reasoning models for reaction.

On the other hand, user tracking with the help of cameras has become stable and has reached a point that the user's skeleton may be tracked with adequate accuracy. The skeleton data that may be generated through Microsoft Kinect or Asus Xtion allow the experimentation with body tracking as well as gesture recognition using hand and the whole body (torso and legs). Moreover, the interaction techniques used for the manipulation of the system should be robust and liberal to possible user behavior that does not match the exact system specifications: the system should be able to prevent reacting in such a way that may be unexpected by the user.

Bryan's system uses a hand-cursor metaphor to provide visual feedback during the interaction of the user with the system regarding the user's actions with their hand. The use of a cursor is crucial especially in the remote handling of intangible interfaces, where the user has no clear picture of how their movement is handled by the system.

The components existing in Bryan's system share common characteristics as far as their handling is concerned. All the interactive elements can be selected in a way that depends on the interaction technique used. Their behavior is more like button style interaction (idle-hover-clicked).

6.1 Natural and full-body interaction

For the real time tracking of the movement of users the service of Kinect Skeleton Tracking is being used. Asus Xtion is the hardware component that provides the camera for

tracking and the software service is build on top of OpenNI [79] modules in C# programming language.

Skeleton Tracking Service keeps track of the skeletons of all the users within range and informs the system only about the nearest one, due to the fact that Bryan is currently single-user. The information that is available contains the user's vital body parts, such as head, torso, hands, legs, feet etc. The information is transmitted to the system and the system translates the users' actions with regard to the current context of use.

The following means of interaction were designed and implemented in order to fit the needs of rich interaction techniques in the demanding field of three dimensional environment interaction. The primary concern in natural user interfaces is the movements and the actions of the users to be similar to the ones they use every day. As a result, the metaphors used include pointing with hands, moving hands in a specific direction (left, right). Finally, the set of interaction techniques are chosen so as to be as less tiring as possible, keeping in mind the fact that extensive employment of gestures for instance may make the system exhausting to use.

6.1.1 User Localization and Skeleton Tracking

The skeleton tracking service all users that are present in the camera's field of view and specified boundaries which are previously defined. Through this the user is tracked and Bryan's system is notified by the service about the following information.

6.1.1.1 The User's position

The user's position according to where his/her torso is in real space is sent. The position then is handled by the Bryan system and if it is the first time that the user is tracked close to the system and they are within the range that the system has defined for user entrance identification, then the system emerges from its idle state end welcomes the user to the system's environment and provides them with a flag menu to select the preferred language that will follow their navigation to the system. On the other hand, if the position of the user is out of range for a predefined time that has been set, then this is the signal for the system to go back to the idle state and restore all the data to the initial state for the next visit of the current or the next user.

6.1.1.2 Hand Tracking

With the extension of his/her hand towards the scene, the user can interact with the system accordingly. The hand's movement is tracked when the user raises his/her hand and

the skeleton tracking service sends the relative to each shoulder position of the hand projected in screen space as well as information about which hand is raised. The 3D point of the hand is used by the system to correspond the virtual hand cursor's position in 3D virtual space with the user's hand in real space. Due to minor instabilities of the hands' detection precision, the system keeps a short queue of the incoming points and uses its average. The queue size is short enough so as not to create irritating latency to the user's actions, but sufficient to keep the cursor steady.

6.1.1.3 Hand Gestures

Through the skeleton tracking service with the implemented gesture generator several gestures are recognized through the movement of skeleton parts points in three dimensions. The generator is used to track hand movement of the user's hands separately as well as connected together.

- The **single hand gestures** include movement of the hands in any direction up, down, right, left, forward and backward. These gestures are used for controlling the multimedia elements view. For example, with the swipe left or right gesture the image show goes to the next or previous image accordingly; with the forward gesture the video pauses and with the backward resumes etc..
- In **both hands gestures** the directions that are used are forward and backward and have the effect of pulling or pushing respectively any item currently in use. These gestures are, for the most part, used for representing movements that are interaction techniques in the system for which the evaluation on how effective Bryan's assisting system is will be conducted – the Timeline [46, 47] which was the Master's thesis entitled "TimeTunnel: Modeling and Interactive Information Visualization Using Three Dimensional Timelines" of Giannis Drossis at HCI Lab of FORTH.

6.1.1.4 Body Gestures

More body gestures that are focalized in leg gestures and the turning of the user's body to the side are used for the same reason as the gestures with both hands mentioned above, that is, to represent interaction techniques of the applications that use Bryan's system. These are placed in the help sub-system in order for the users to be trained on how to interact with their system. All the complex interaction techniques that are used in the applications should be passed through animations in Bryan's system so that the virtual

human can show them to the user, explain their role in the interaction field and teach their use to the users by performing them.

6.1.1.5 Interaction with Mobile Devices

In order to get real time data of a smart phone, the Harvester mobile application was created in the context of the Master's thesis of Ftylitakis Nikolaos entitled "Smartphone exploitation in Ambient Intelligence Environments" at HCI Lab of FORTH. This application provides gesture recognition for a variety of gestures. Recognition is achieved through the sensors of the mobile device. The gesture vocabulary of the application includes the following gestures:

- *Cover*: It is generated when the mobile device is facing upwards and an object is placed over the mobile phone at a maximum distance of 4cm.
- *Hover*: as with Cover gesture, it is produced when the mobile device is facing upwards and an object is placed over the mobile phone. The difference is that, in this case, the object has to remain above the device for 3 seconds before the Hover gesture is generated.
- *Double Tap*: A quick double tap anywhere on the device.
- *Pick Up:* The mobile phone is facing upwards and is left still on a surface. The user grabs the device and, with a quick movement, brings it to vertical position.
- *Shake:* The intense movement of the device back and forth, up and down or right and left for 3 times in a very short period of time. The gesture includes the axis of movement.
- *Turn Over:* Once the device is faced downwards and is close to an object, it is considered as turned over. For example, a mobile phone is placed on a table facing downwards.
- *Twist*: The device has an initial orientation. A quick change of the orientation to another value and then back again to the initial orientation constitutes a twist. The intermediate orientation is used to extract the direction of the gesture. The direction values are the Front, Back, Left and Right twists. For example a mobile phone's top is faced upwards. The user twists left, rotating the device 90 degrees counterclockwise for one second and then restoring the orientation to its initial value. This is a Twist gesture with left direction.
- *Swipe:* A quick movement of the device over an axis. Includes the direction of the movement in the 3D space.

The gestures used for Bryan's system are the following:

- Hover, which is used for activating the interaction with the mobile device.
- Turn Over, which is used for deactivating the interaction with the mobile device.
- Twist, which is used for controlling the view of multimedia components. For example, when the scene is in video mode projection, then in order for the video to be paused, the user enables the twist forward gesture whereas to be resumed the twist backward is employed.

However, the additional gestures may be added in a straightforward manner by mapping the gesture with the preferred action that will be executed when the gesture is performed. For example, if a developer wants to manipulate the view of the three dimensional models with the swipe gesture, then he/she just maps the gesture with the action like "swipe-left ->3D model-left". The mapping is loaded into the data retrieval process and constructs the structures needed. In the Harvester's case, only the gestures involved in the mapping are used and the others are ignored.

In order for a mobile device to make use of the Harvester application, the application should firstly be installed, a handler should run with a context name for getting the events sent in another pc and then the application should be run through the mobile device with the handler's IP to connect with. Whenever a gesture is performed by the mobile device then the handler gets the appropriate event and through middleware events sends it to the system that listens to those events and manages them.

6.1.1.6 Verbal Interaction

With the help of the Signal Processing Laboratory (SPL) of ICS FORTH and especially Elena Karamichali, a speech recognition system was created based on the HTK Speech Recognition Toolkit and ATK Software. [74]. This system is in an experimental stage so its use is limited. It recognizes combination of words ranging from 2 to 3 words. It does not support single word recognition or monosyllabic words yet. The use of 2 or 3 words operates more precisely because it restricts the possible combinations. The more complex a word, the larger the percentage of recognition success is.

Because of the fact that the word combination is limited and since we wished the system to have a primary vocal interaction, the decision was to create a vocabulary of spoken commands that would be given from the user and recognized through the system with the help of the speech recognition system. These commands would only be used for components manipulation, for instance, to change language, enable or disable mute etc. and not for conversation with the virtual human, which would be the ideal for such a system.

The process followed in order for the system to be able to get vocal commands and understand them is as follows:

1. A group of commands (30) suitable for specific actions to the system was compiled. Some indicative commands include “Go Right”, “Go Left”, “Start Speaking”, “Stop Speaking”, “Main Menu”, “Enlarge Display”, “Change Language”.
2. The commands were checked to see if they were appropriate for the recognition system. For example, one word commands were rejected and others with low percentage of recognition success were changed.
3. A group of ten people to record the commands was selected. The group was composed of as many different voices as possible so that a wide range of sounds and tones would be available. Each person recorded each command loud and clear repeatedly (10 times) standing in a specific distance from the ASUS XTION microphones.
4. All the recordings were passed through the speech recognition algorithm so as to train the system to recognize the commands uttered by any user.
5. Then, a test was carried out with the participation of both people who were present in the recording process, as well as people that were not and the results were satisfactory.
6. Finally, a communication protocol was established through middleware events: when a spoken command is recognized, an event is sent so as to inform Bryan’s system and prompt it to react appropriately.

Unfortunately, to the present, the recording and the commands are in Greek only. Because the accent of the Greeks is quite different from the British there was a problem with the existing system in English recognition, but we hope that this obstacle will be overcome in the future. What is of utmost importance is the fact that all the commands given by the ten members of the group were recognized as far as they were uttered clearly.

6.2 Communication with the environment

The system has been developed and will be exhibited in a building dedicated to Ambient Intelligence within the premises of the Institute of Computer Science (ICS) of the Foundation for Research and Technology (FORTH). Communication with the applications used for input from the environment, such as Skeleton Tracking Service, Speech Recognition and Harvester is supported by a middleware layer based on CORBA, developed for the intercommunication of applications and services [middleware]. The middleware layer offers tools and libraries that allow the creation of services using Application Programming Interfaces (APIs) in any of the supported programming languages, C++, .NET languages, Python and Java. The services running using the middleware may be spread across the network and programmers do not interfere with network connection establishment. On the contrary, developers focus only on the functionality that each service should offer, implementing the desired interfaces as a service and creating wrapper classes that can handle incoming data for any client that receives them.

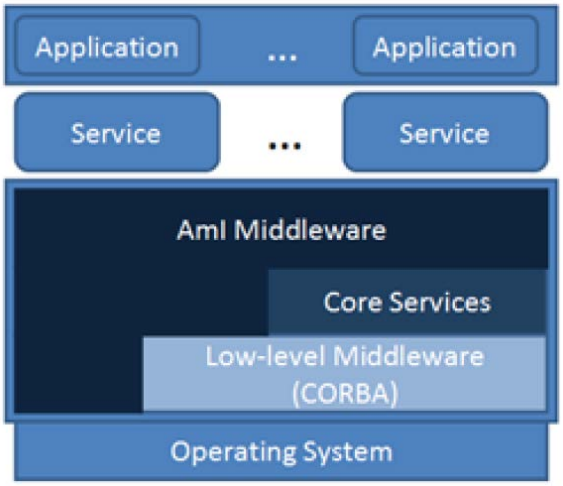


Fig. 58: The architecture of AmI Middleware

7 Implementation

The system is written in C# programming language on Unity Platform [80], which is a cross platform game engine with a built-in IDE (Integrated Development Environment) developed by Unity Technologies. Unity supports deployment to multiple platforms. Within a project, developers have control over delivery to mobile devices, web browsers, desktops, consoles and allow specification of texture compression and resolution settings for each supported platform. It currently provides development for iOS, Android, Windows, Linux,

Flash, web browsers and others. The platform supports the developers with features such as rendering, scripting, asset tracking, platforms and physics.

7.1 Information Architecture

The components that will be described in the following segments compose several types of information which may be used in a variety of ways, according to the purpose of their use. The system is represented by Unity GameObjects which contain two main parts that manipulate different aspects of an element:

The Component part, which contains the necessary information for each element and implements all the needed functions in order to handle it at a higher level. Each component may be part of a hierarchy, but is able to render itself and contains values for:

- Transformation (translation, rotation, scale)
- Rendering (visibility, transparency, shaders)
- Physics (bounding box, collider)

The Script part, which defines the behavior of each element and controls every component associated with. The scripts are attached as components to other components or GameObjects in Unity.

Scripting architecture is adopted aiming at keeping each element's distinct implementation from its behavior. This way, code reusability is achieved and system's extendibility is assured. All components and scripts are registered to the system at startup or at runtime and the engine of the system is responsible for updating, positioning (in scene and in hierarchy) and drawing them. A hierarchical mechanism is implemented in order to allow parent-child relationship. Components which are children of GameObjects or other components use a local transformation with regard to their parents.

7.2 Basic Visualization Components

For the scene provision some existing **unity components** are used which setup the entire scene through their composition and organization in other imported models. The components used are the following:

Asset Components are the models, textures, sounds and all the "content" files of which the game is composed. Asset components include, but are not limited to, texture2D, materials, movie texture, fonts and audio clips.

Asset components are used in the information visualization system of Bryan. The information can be represented through images, videos, text and audio. Images for their graphical view in the Unity scene are texture2D items which are rendered through materials in various surfaces as projection screen's sheet. Videos use movie textures for their representation and they are viewed through a second material in the projection screen sheet. Shaders are used on the Texture2D to provide special effects, such as transparency, in rendering. Text components are represented with the help of fonts which are viewed in the scene through materials. Audio clips containing the audio data used in audio sources are used for the audio of the scene. They can be manipulated through code to start, stop and pause or change their volume.

The Mechanim Animation System (M.A.S.) is a tool provided by Unity in order to handle animations. The M.A.S. can be split in three major stages: 1) the asset preparation and import which is done by artists or animators with 3rd party tools such as Max or Maya and is independent of the M.A.S. features, 2) the character setup which can be either humanoid or generic setup and 3) the stage where characters come to life, which involves the animation clips setup, interactions between animations, state machines, animation parameters animator controllers, transitions and blend trees.

As part of this work the M.A.S. is used to provide animation control for every type of animated model that is contained in the system, such as the virtual human and the projection screen. Animator controllers, animation layers, state machines and sub-state machines inside the animator controllers and parameters that manage the states have been created with a view to achieve this control. Animation blending and transitions from one animation to the other are the results and can be previewed before the final version through

the M.A.S. editor. Different animator controllers are provided for each 3D animated model because each model has its own animations, state machine and conditions for the transitioning. Two layers are used in the virtual human animator controller, one for controlling the head animations and one for the body animations. These animation layers are executed simultaneously in order to support speak and body movement at the same time. (Fig. 59) All the interactive button-style components use a generic animator controller with the same state machine but with different animations which change correspondingly. The states of the state machine are idle, hover and click. An integer parameter controls the state and the animation transition every time a change is needed. (Fig. 60). Animator controllers are supported for the hand cursor (Fig. 62) , for each category (Fig. 61) and for the projection screen (Fig. 63) as well.

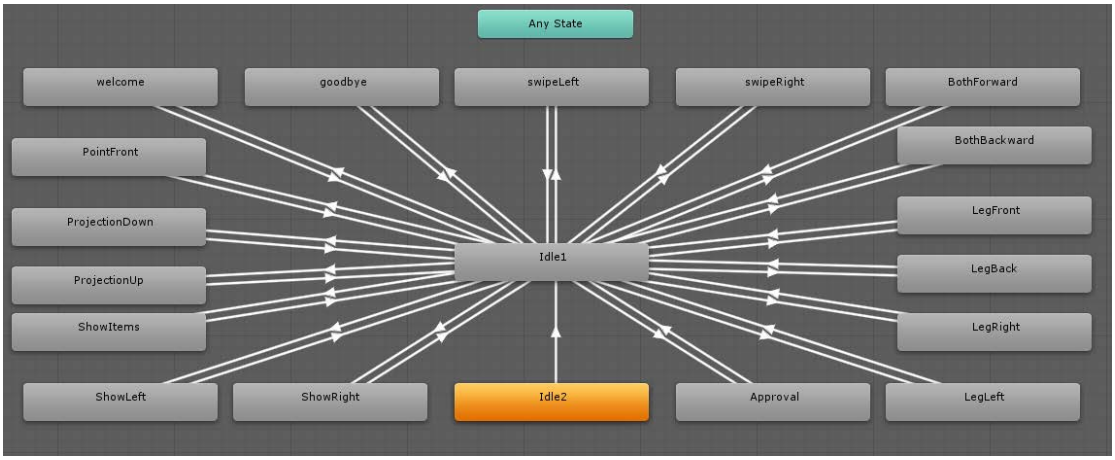


Fig. 59 : Animator Controller of Bryan's Gestures

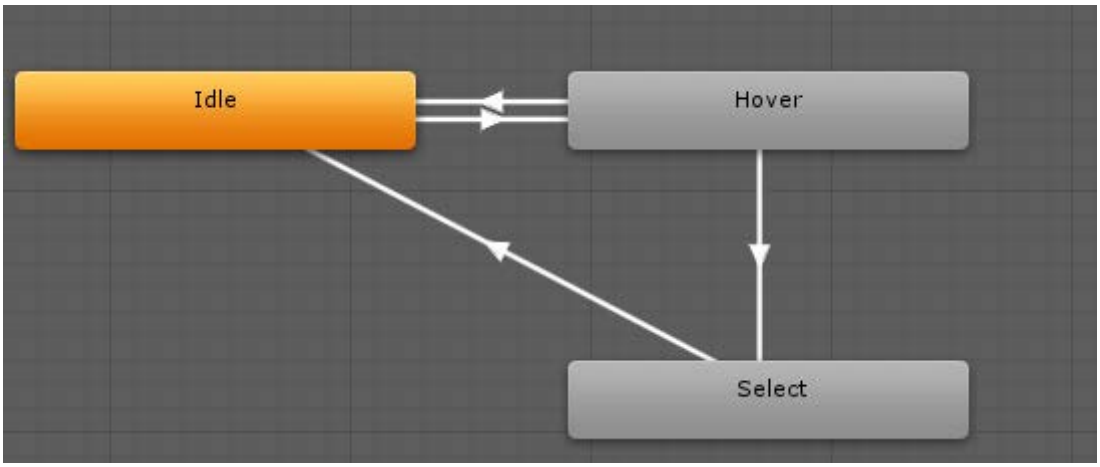


Fig. 60: Animator Controller of button style elements

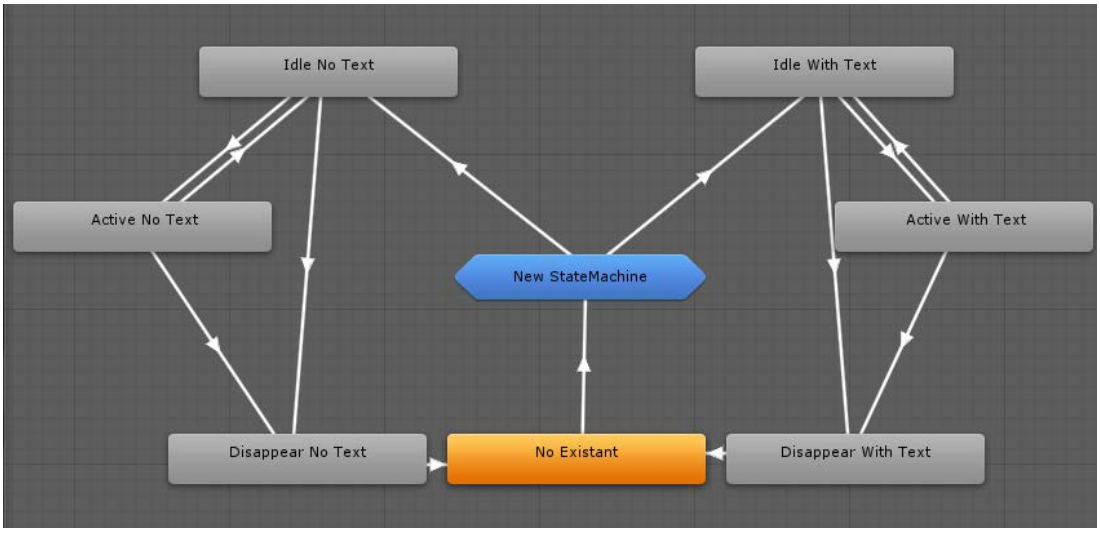


Fig. 61 : Category Animator Controller

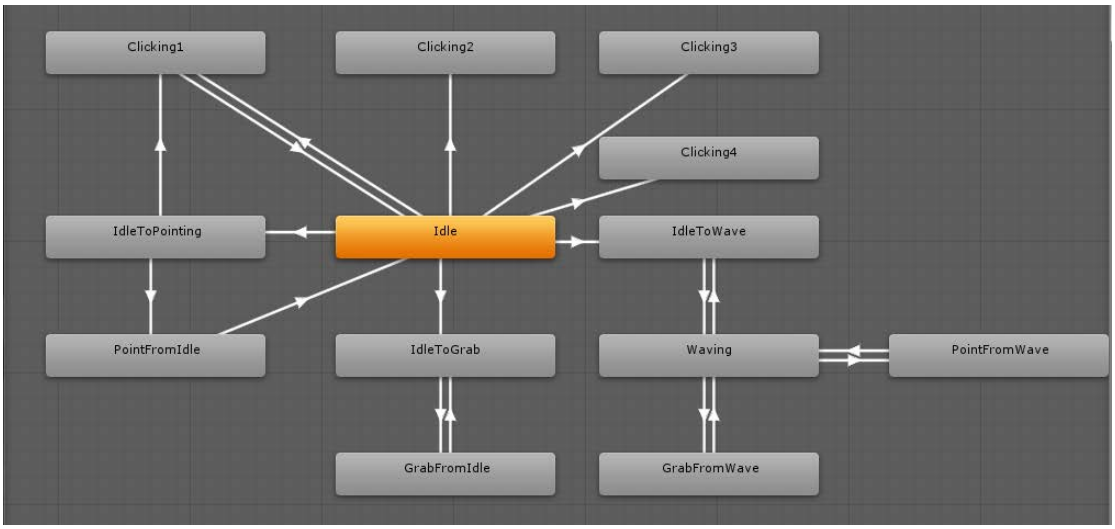


Fig. 62 : Virtual Hand-Cursor Animator Controller

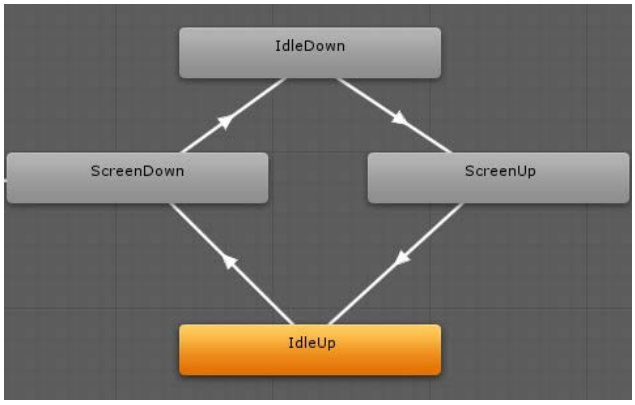


Fig. 63 : Projection Screen Controller

Physics components allow the developer to provide realistic motion in objects and reaction to collisions. Every selectable component should have its own collider in order to be raycastable from screen space to 3d space and to be able to respond accordingly in collision detection. Ray-casting is a common technique used for ray-surface intersection. The colliders can have various shapes such as capsular, spherical, cylindrical, cubic and generally polygonal. Each collider has its center and size.

Mesh Components are the main 3D graphics primitive of Unity. These components contain **mesh filters** which are the mesh intermediates from assets to the mesh renderers; **mesh renderers** take the geometry from the mesh filter and render it at the position defined by the object Transform component; **skinned mesh renderers** are automatically added to imported skinned meshes, and **text meshes** generate geometry that displays text strings.

Mesh filters and consequently mesh renderers are part of every component that is going to be rendered in the scene. The renderers can support multiple materials as in the projection screen sheet where images and videos can be rendered just through alternation of the two materials. All the skinned models of the scene contain a skinned mesh renderer which encloses the materials, bounds and the mesh described by it. All those properties can change at runtime. Text meshes are used for every text rendered in the scene such as titles in menus. The properties of a text mesh are text, offset Z, character size, line spacing, anchor, alignment, tab size and font, font size and style. All the properties are dynamic and changeable at runtime if needed.

Audio Components implement sound in Unity. These components contain the audio listener, which when added to a camera provides 3D position sound, and the audio source, which when added to a GameObject makes it play sound. The Audio Listener acts as a microphone-like device. It receives input from any given audio source in the scene and plays sounds through the computer speakers. The Audio Source plays back an audio clip in the scene. It has several changeable properties such as mute, loop, priority, volume, and pitch.

Rendering components contain all the components related to rendering in game and user interface elements, lighting and special effects. Cameras are the elements that capture and display the virtual world to the user. The scene can have an unlimited number of cameras which will be set to render in any order, at any place of the scene or only certain parts of the screen. Cameras permit further customization and manipulation in order for the system to be enhanced.

In Bryan's system, the Main Camera is the only camera of the scene and controls the user's view. Its transformation changes on demand and a mouse manipulator is responsible for its movement in the virtual world. On camera's movement (animation) audio effects are enabled giving a more pleasant experience to the user. The camera's projection is perspective and the field of view is changeable at runtime as needed. A GUI (Graphical User Interface) Layer is attached in the camera in order to render the GUI text which is used to print text to screen in 2D. Subtitles are drawn through GUI Labels which use GUI text. Lights illuminate the scene and the objects in order to create the perfect visual ambience. There are several lights in the virtual scene, some of which are static and remain in the same position, whereas some others are attached to the camera or other moving objects and move accordingly. The lights are either point or directional. Additionally, a light-mapping has been created for the room overview in order for the room walls, roof, floor and wallpapers to be lightened and shadowed in a realistic way.

The **Transform Component** is applied for components that handle object positioning outside of physics. The Transform component determines the position, rotation and scale of each object in the scene and every object placed in the scene has its own Transform.

GameObjects are containers for all other components. All the objects in a Unity game are GameObjects that contain different components. GameObjects, apart from components, have a Tag, a Layer and a Name. The tag is used for quick search of objects and layers can be used to cast rays, render, or apply lighting to certain groups of objects only. Therefore all the scene objects in the system are GameObjects which contain components with functionality.

The **Unity GUI Group** is the GUI creation system built into Unity which creates different controls and defines the content and appearance of those controls. This group contains GUI skins and GUI styles, which are collections of custom attributes for use with Unity GUI. Each control has its own style definition. GUI skin is a collection of GUI styles that can be applied to the GUI. Skins allow applying style to an entire user interface, instead of a single control. For the subtitles GUI text, the label text color and the background were changed and a style was created for manipulating the alignment, font size, word wrapping of the text.

Build-in or new Shaders. Unity is supplied with more than eighty built in shaders, which essentially combine shader code with parameters like textures. Shaders contain code that defines the type of properties and assets to be used. Materials allow adjusting properties and assigning assets. Some of the shaders that are used in materials for the system are

transparent cutout bumped diffuse, self illuminated bumped diffuse, unlit texture, diffuse, vertex colored and legacy shaders lightmapped bumped diffuse. Because by using 3D text the default shader was the same with the GUI text and appeared always on top of everything, a new shader was created for the text rendering to behave properly, the 3D Text shader.

Apart from the Unity components that exist in the Unity platform, some additional models were created outside the Unity, using the Autodesk Softimage 3D computer graphics application, with the help of the graphic artist, Mr. Antonis Katzourakis, and imported into the Unity's platform to enhance the information visualization and the functionality of the system. Those models are:

a. The Virtual Human –Bryan
b. The Projection Screen
c. The Hand Cursor
d. The Virtual Room

All of the above imported models contain Transform, Animator and Animator Controller, Mesh Renderer or Skinned Mesh Renderer, Collider if needed, materials and shaders for rendering textures and Lights.

7.3 Core Engine Implementation Details

The Bryan system architecture consists of two main parts, scripts and services. The scripts are individual components which may have many instances and can be applied to other components or GameObjects, whereas services are unique elements which offer specific functionality.

7.3.1 Services

Services are singleton classes which either offer functionality that may be accessed by the scripts at runtime or propagate information which is handled by the scripts. The singleton pattern has been chosen due to the fact that services are unique and therefore their instantiation should be limited to one object at a time. Their role is to help the system to communicate with the environment and the user through input and output events. Input can be defined as events coming from sensors and voice recognition and output as the speech of the virtual human, as shown in the table below. For every service there is one server and one listener, on the server's perspective the recognition of the actions is

performed and messages are sent through Famine Middleware functions [6]. The listener receives the evens coming from the middleware and reacts accordingly.

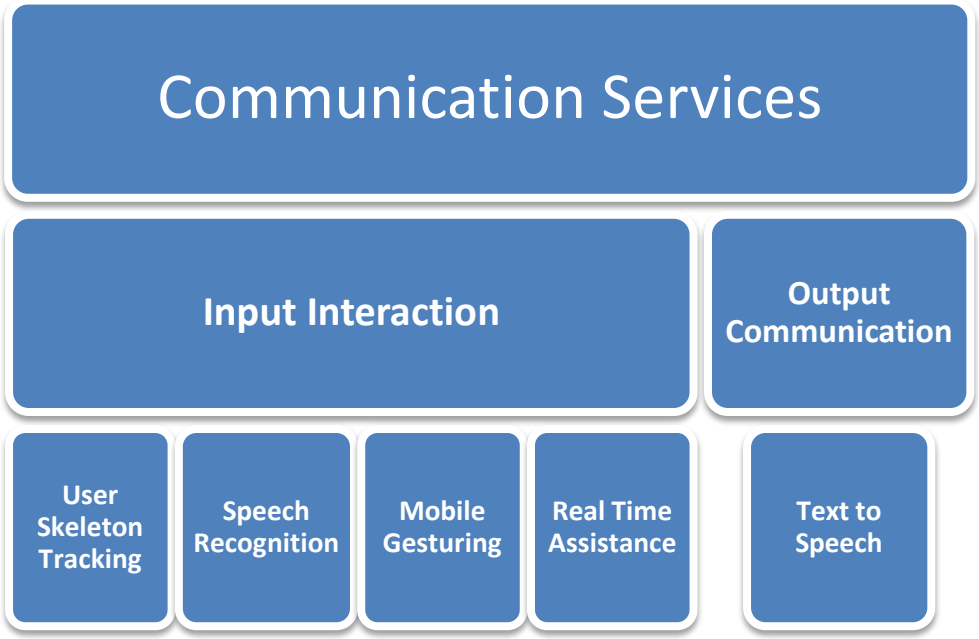


Diagram 7 : The services available inside the Bryan' System.

7.3.1.1 Input Communication Services

The input communication services are events that are raised from the environment, coming from the user through his/her voice or body gestures or using mobile devices and derived as real time commands from other applications, usually the ones which use the system as an assisting tool. The description of those services is as follows.

- **User Skeleton Tracking**

This service performs as a type of primary interaction with the system. It provides recognition of the user's position in the environment, the user's hand position as well as specific gestures that the user can make in order to interact with the system.

Hand tracking, with either the left or right hand, allows the user to manipulate the virtual hand cursor of the system by creating a position mapping from real space and the user's hand to virtual 3D space and the virtual cursor. This component enables the user to select items in the scene and obtain more information, as well as to manipulate the

information visualization. This can be achieved because both the virtual cursor and the selectable items contain bounding boxes each fitted to their shape and because events are raised and inform the system to react accordingly through callback functions in collision detection situations between them. The service for the hand tracking provides some event functions through Famine Middleware that gives the input position such as:

void *Event*_**HandPointsAt**(**in** float x,**in** float y,**in** float z, **in** boolean isLeft);

void *Event*_**HandPointsAt_ScreenCoords**(**in** float x,**in** float y, **in** boolean isLeft);

Apart from hand tracking, several **hand** and **body gestures** are recognized during the movements of the user. Those gestures are used by the system to provide interaction to the user with it. Those gestures are used as a secondary interaction technique to enhance user experience. Some examples of gestures are swipe left, right, forward and backward which are used to control information visualization. Swipe left and right are used for navigating to previous or next items whereas forward and backward for playing or pausing the video. There is also a wave gesture to which the assistant responds with a greeting back. Those gestures are achieved through these event functions of Famine middleware:

void *Event*_**HandGestureTracked**(**in** GestureDirection d, **in** boolean isLeft);

void *Event*_**HandWave**(**in** float x,**in** float y,**in** float z);

For the needs of Evaluation and because the TimeViewer [46] system uses such gestures, gestures performed with both hands and legs for navigation to the system of Timeline were also used. In addition, the torso position was used in order to understand whether a user was in the defined area for interaction with the Bryan system or not. These event functions are:

void *Event*_**BothHandsGestureTracked**(**in** GestureDirection d);

void *Event*_**LegMoved**(**in** Direction d, **in** boolean isLeft);

void *Event*_**TorsoMoved**(**in** float x,**in** float y,**in** float z);

- **Speech Recognition**

As long as a virtual human is the main means of giving information to the user and users are those who interact with the system, a more human-like communication style is provided through speech recognition. A mechanism for speech recognition created by the Signal

Processing Lab of FORTH is used, which, through Famine events, informs the Bryan system of what was spoken and if this sequence of words is in the set of the sentences that the system is interested in. For the purposes of this thesis and because the speech recognition mechanism is still in the experimental stage, the set of recognizable sentences is narrow. Speech recognition is mainly used for voice commands of two or three words coming from the user in order for him/her to interact with the system and be able to manipulate it with interactive button-style items. The event function used from Famine middleware is:

void *Event*_**SpeechRecognition**(**in** long text);

and when the listener – in this case the system of Bryan – receives the events coming matches the text coming with the specified commands and reacts accordingly if the text is in the predefined set. The speech recognition is used as an auxiliary means of interaction techniques.

- **Mobile gesturing**

The user can interact with the system and control the information displayed through mobile device gestures. A mobile gesture generator was implemented to enable the user to communicate with the system with the aid of his/her phone. Through sensors such as accelerometer, gyroscope, compass and magnetometer several techniques of interaction were created based on mobile devices. Two services were created in order for a system to recognize those techniques. The first refers to the server which connects the mobile device with the communication environment and enables it to send gestures. The latter is the one responsible for starting the recognition process, checking if the device is connected, and for the type of gestures recognized and sent. Some of the gestures used are cover, hover and twist in different directions. The (event) functions provided are:

void *Event*_**DeviceConnected** (**in** string contextName);

void *Event*_**DeviceDisconnected**(**in** string contextName);

void **StartGestureRecognizer**(**in** Gesture gestureId);

void **StopGestureRecognizer**(**in** Gesture gestureId);

void *Event*_**GestureRecognized**(**in** Gesture gestureId , **in** Direction direction);

void *Event*_**DeviceConnected**(**in** Device newDevice);

- **Real Time Assistance**

The purpose of this functionality is to give an assisting tool to the applications so they can provide the user with real time assistance through the use of the virtual human and information spoken and displayed in the scene. This form of help can be provided at any point of the system runtime and also, when the application deems that the user needs assistance. The assistance can be presented through speech in many languages, animations video, images, text or all of them in combination. The application can also change the language of the system on demand. The functions used in order for the assistance to be enabled are the following:

void **_Speak**(**in** wstring speech); void **_Animate**(**in** string animation);

void **_VideoShow**(**in** string name); void **_ImageShow**(**in** string name);

void **_TextShow**(**in** string name); void **_SetLanguage**(**in** string language);

7.3.1.2 Output communication Services

The output communication services are used to provide communication between the user and the system and they come from the system.

- **Text to Speech**

This service is created in order to give voice to the assistant and speak/behave in a human way and through this to communicate all the information and descriptions of the system to the user. Communication is not limited to speech as facial expressions were also created in order for the virtual human to be more realistic and believable as humanoid. All functions in this service are performed at runtime. Those functions are:

void **_Speak**(**in** wstring tts); void **_Stop**(); void **_Pause**();

void **_Resume**(); void **SetLanguage**(in string language);

void **SetGender**(**in** long gender); void **SetRate**(**in** long rate);

void **SetVolume**(**in** long volume);

The functions provided are related to what the assistant can say, in what way, which voice to use, and in which language to speak. In addition, the speech can be controlled by pausing or stopping or resuming the speaking.

Apart from the functions related to voice, there are some event functions that refer to the facial expressions and mouth movements created by the spoken text. These functions provide information about the phonemes and visemes that a text to speech can produce and are the following:

void *Event*_**GetCurrentViseme**(**in** VisemeInfo currentViseme);

void *Event*_**GetVisemes**(**in** VisemeSeq visemes, **in** long visemes_count);

void *Event*_**GetCurrentPhoneme**(**in** PhonemeInfo currentPhoneme);

void *Event*_**GetPhonemes**(**in** PhonemeSeq phonemes, **in** long phonemes_count);

Viseme information includes viseme audio position, viseme duration, viseme emphasis, the next viseme and the current viseme. The phoneme information contains phoneme audio position, phoneme duration, phoneme emphasis, the next phoneme and the current one. The emphasis can be normal, stressed or emphasized. Additional available functions are included to retrieve the available languages installed in a system and to raise an event upon starting and finishing speaking.

voices **GetAvailableVoices**();

void *Event*_**GetStartOfStream**(**in** boolean start);

void *Event*_**GetEndOfStream**(**in** boolean end);

7.3.2 Scripts

The scripts implemented in the Bryan System refer to items that may have multiple instances and are attached to other Components. The main script is the one that manages the states of the system as a whole and is responsible for informing individual components or GameObjects about changes. The figure below shows an overview of the script hierarchy. All the system script parts will be described thoroughly in the following sections.

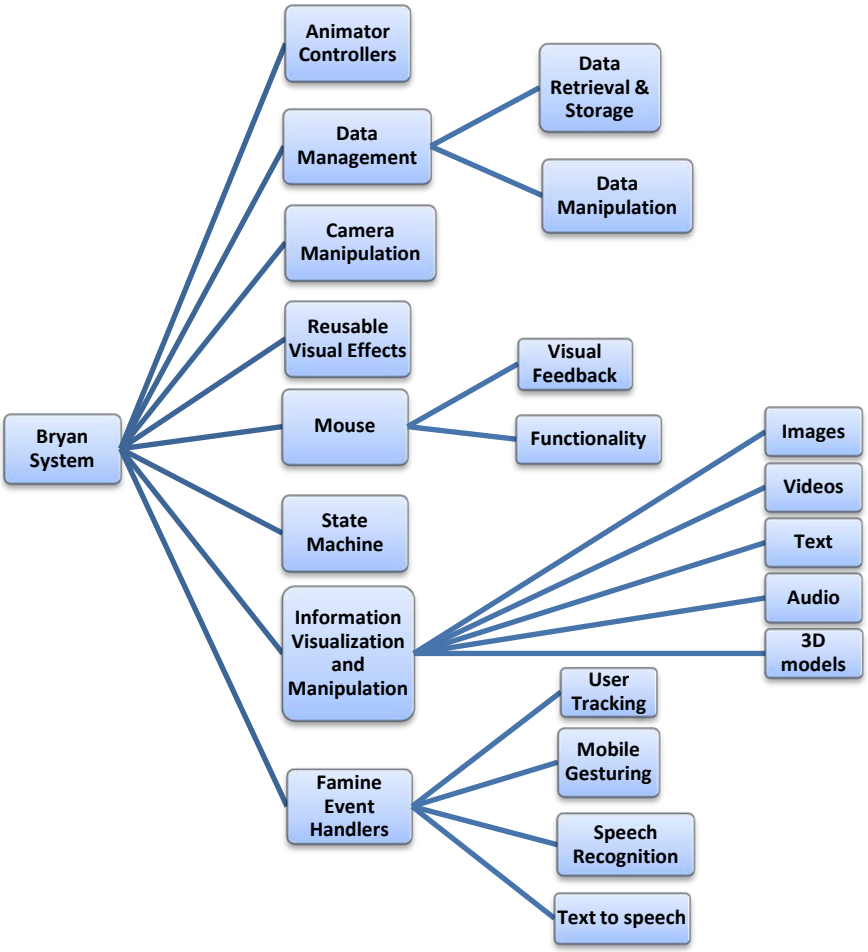


Diagram 8 : The scripts architecture of Bryan System

7.3.2.1 Animator Controllers

In order to control the animations and animators of the Unity mechanim system and to manipulate their states and parameters at runtime, one script is created for each animator controller. Each script contains three **callback** functions and one for **update**. The callbacks are used in order be informed of interaction actions that are related to hand-tracking and the movement of virtual hand cursor. The callbacks that are used are raised when the cursor is over an item, when the cursor stops being over it and when the cursor selects it. Such elements are all the button-style items as previous, next, play, pause, mute etc., the language items and the selectable imported animated models.

According to the cursor's state, all the interactive animated objects of the scene can be in one of three different states, idle, hover or clickable. While the cursor is not intersecting any element, the hand cursor is in idle state and plays the idle animation while the remaining elements are in their default idle state and animation. When moving the

cursor over a selectable item, an event (hover) is sent to the Hover Callback function which indicates animation change for the item, as well as for the virtual cursor. After a specified timeout over an interactive component, the clickable state is activated, an event is raised and the Click Callback is called to change the animation and the state of the item, as well as to perform the corresponding action that is intended to. Finally, the change of animations and the selection is takes place in the update function.

The components of the scene that have animator controllers are the hand cursor, all the button-style selectable objects, the information visualization elements (image, video, text, 3d models, audio) and the category and language containers. Whenever a component is added to the scene, its animator controller is created and is attached to the component and the callback actions are registered to listen for events coming from the cursor (idle, hover, click).

The virtual assistant has an animator controller in order to speak, show facial expressions (e.g. happy or angry) and to perform body animations. The controller contains an update function for changing the animation state machine when needed. So as to apply animation transitions, a function is responsible for the mapping of phonemes to visemes coming from the text to speech service and playing the corresponding animation. The assistant's body and facial animations can be performed simultaneously, using the layer system of Unity mechanim. Two different layers along with state machines for the assistant's animations are created, one for the Visemes' representation, in which the Visemes are inter-connected for smoother lip synchronization and one for the character's body movements to blend in.

7.3.2.2 Camera Manipulator

In order to change the view of the virtual world, a camera manipulator was created that allows the management of the camera's position, orientation and field of view. Apart from simply moving the camera in the virtual world, camera manipulator also offers the ability to create moving paths through key frame creation using the Unity Animation editor. Finally, camera manipulator can also play audio files while moving the camera.

7.3.2.3 Reusable Visual Effects

The visual effects were created for enhancing the user experience and making the functionality provided more appealing. Every GameObject can contain those effects until it corresponds to each effect's requirements. The effects are attached to the GameObjects dynamically and can be removed or deactivated at any time.

The effects provided are the following:

- *Fade in and Fade out*

This effect can be used in all the components that have materials and use a shader supporting transparency. This effect is used so that any item can smoothly appear or disappear from the scene in order for the transition to be soft and the user not to be surprised when an item is appears in the virtual environment. The technique that is used for the achievement of this effect is the gradual increasing or decreasing of the alpha value of each material (texture or text) of a component and of its children's materials.

- *Interpolation*

This effect is aimed to provide transformation from one position to another of a GameObject. Interpolation is linear, using Vector3 Lerp function which takes the start point and the end point of an object and with time parameterization performs the transformation. By changing the timing parameters, the transition from one point to another appears smooth.

- *Rotation*

This effect is used to provide rotation from one orientation to another using linear interpolation. Interpolation is linear, using Quaternion Lerp function which takes the start point and the end point of an object and with time parameterization performs the transformation. By changing the timing parameters, the transition from one point to another appears smooth.

- *Resize*

For rescaling any component needed a resizing technique was made in order to scale up or down the objects of the scene. The alternation is proportional to all axes by multiplying them with the scale factor for increasing or decreasing their size. The scale factor should be non zero and if it is equal to zero then that means that the object is not visible anymore.

- *Change Material*

This effect is used for changing materials and applying new textures or movie textures. Upon changing the texture applied, the Update function is responsible for updating the rendered texture; either it is a video or a sprite.

- *Show-Hide items*

  This effect affects the visibility of a GameObject in the scene. It takes the renderer of the component and its children's renderers as well, and either enables or disables them according to the game state and situation on the interaction. This effect is used for instantly disappear or appear items on demand.

7.3.2.4 Mouse

The virtual hand cursor which is used to replace the mouse in the scene is divided into two scripts. The first script is responsible for the visual feedback of its movement, while the second one is in control of interacting with scene components.

As far as the visual feedback is concerned, the hand cursor is designed so as to contain a virtual clock on the top of the hand which upon moving over a selectable object starts turning green and when it finishes it raises an event for clicking. The clock is enabled when the mouse moves over other elements and is disabled upon leaving. Furthermore, through the animator controller containing different animations on hover, idle and select state, the animations played are pointing, open hand and clicking respectively.

The movement of the hand cursor is acts as a visual feedback in the scene. A script has been created in order to display the movement that a user executes during his/her interaction with the system. A mapping is created through projecting the actual position of the hand in world space (given through Famine middleware events) to the position of the virtual hand cursor in the virtual world.

The functionality of the hand cursor is the same traditional functionality of a default mouse, differentiated by time-based selection instead of clicking. The corresponding callback of the selected item is called through the states that the mouse changes. Sometimes through a hand click the game's entire state can change because it selects an entire section or to return to previous state and consequently destroy components that are not used anymore.

7.3.2.5 Event Handlers

Those handlers are scripts that take the role of listeners in Famine middleware events and when an event is raised, the appropriate functionality is performed. In the scripting part of implementation there are five handlers as the services provided. The Skeleton Tracking

handler, the Mobile Gesturing handler, the Speech Recognition handler and the Text To Speech (TTS) handler.

7.3.2.6 Data Management

In order to deal with all the content of the system a mechanism of data management was developed which is divided in two parts: firstly the data **retrieval** from XMLs and the **storage** in Serializable classes and secondly the **manipulation** of the stored data through "loaders", "creators" and "updaters" of the content according to interaction actions of the user, the system and the environment.

- Data Retrieval & Storage

All the content is edited by the developers in XML files, organized in an understandable terminology of what every element represents. All the configurable classes are serializable, i.e. they are converted into reasonable forms that can be readily transported. The most important functions of the serialization process are Serialize and Deserialize. With the Serialize the objects are transformed into XML files, whereas through Deserialize the reconstruction of the objects in the storage classes is performed. A XML Writer is used for creating and filling the XML file with the accompanying data during Serialization, while a XML reader is created so as to read the content data created from the developer through the Deserialize class. The serialization does not include type information, only public fields and properties of an object. Those fields can be single C# types like strings and floats or enumerations and data structures like lists. A script for dictionary serialization was also made.

The data retrieved from each XML are stored correspondingly in classes that were created for this purpose. All information in the system has the need of grouping its data. To achieve this, a data class is created for better management of the information for each sub-system of the Bryan's environment. The main informative subsystems are the Training, the Categories and the Tutorial. Each of the above is responsible for a large amount of data and their information visualization.

Tutorial information is structured in chapters. Each chapter is a tutorial collection class which contains a dictionary with the title of the chapter and a list of tutorial data per entry. Tutorial data class contains (a) a dictionary for mapping the description with the corresponding language, (b) the representative animation of the assistant for each

information item, (c) the duration of the displayed item, (d) the path to the folder where to retrieve the displayed element and (e) the type of the element.

Training information contains a list with all the interaction techniques which the users can be taught. The Training collection class contains a Train data. A Train data contains a dictionary of animations and a mapping of the description for the associated animation interaction technique for each language. It also contains the number of tries that the user has in order to accomplish the interaction method.

Categories collection class is structured in a list of Category data, each one describing one type of information. Each category data contains (a) a dictionary with names of the category per language, (b) the folders to retrieve the data for images, videos, audio, text and 3d models, (c) the representative category item which can be a 3d model or image, (d) the path to the representative item and (e) the introductory description spoken for every available language. It also holds dictionaries for the each information type visualized in the scene (image, video, audio, text, 3d model). Each of those dictionaries maps the information item with spoken description in the corresponding languages. An extra dictionary is created in order to retrieve the path of each type of information.

Some minor Serializable collections are the TTS, Language, Category Grid, Mobile handler, Video and Image classes. They hold general information for the environment setup and the information. TTS class contains the gender, volume and the rate of the assistant's voice. The Language class contains a dictionary for mapping the flags to their language, the folder path where to retrieve the flags and a list with the available languages. The Grid holds the number of categories and the presentation format that they will be displayed, spherically, horizontally, vertically or in carousel style. The Video and Image classes contain the available formats that can be used in the system for images and videos. Finally, the Mobile handler class contains a list of the gestures coming from the mobile device in which the system will listen to.

- Data Manipulation

This part of content loading refers to the real time creation of the scene's main data collections and the simultaneous creation of their update machines according to the changes needed for the interaction with the user or the environment. The main collections are, as mentioned above, the Tutorial, Category and Training and their information manipulation.

As far as the Tutorial data manipulation is concerned, a Tutorial container is initially created and filled with the needed components, a chapter and an information displayed controller. All the data stored in the Tutorial data class are available to be loaded at any time and displayed wherever needed. Using these data, a chapter list is structured and included in the chapter container. The information is displayed sequentially on the projection screen and according to the chapter's structure, unless the user interferes through verbal or non-verbal commands. Supplementary information is presented by the virtual character using gestures and speech which is also retrieved accordingly from the Tutorial data collection.

The training data manipulation is similar to the tutorial one: a Train container is created and all the necessary components are added. The components used are a container for displaying the animation list and a controller script for navigation. The animations for the help of the interaction methods are loaded from the Training data and are visualized through animations performed by the virtual human and the description read aloud using synthetic speech.

The categories data manipulation is structured in a grid which contains all the categories and the accompanying information stored in the category data collection. Each category is hosted in a cylindrical three dimensional animated model, displaying its title along with its representative item. Their information is filtered by its type (e.g. image, text etc.), retrieved and created from the categories data at runtime when a category is selected. Upon the selection of a type of information, the content which corresponds to the selected one is loaded from the data collection and is visualized appropriately, either in the projection screen or in space. Similarly to other components in the scene, information manipulation can be applied from the user through controllers that are provided, either with verbal or non verbal interaction techniques.

The Language is a data manipulation formed in a similar way with the main collections. It contains all the available languages, along with their flags, according to the data collection containers. It consists of a selectable grid, which can be expanded in order to display additional items initially hidden. The selected language can be changed at runtime, resulting in the automatic update of all the content to match the new language.

7.3.2.7 Global State Machine

Global state machine is the state machine holds the system's status. It is divided in three main states: idle, language menu state and information menu state. The information menu in its turn is separated in the three main information states which are the tutorial, training and categories. The Categories state has one sub-state, the multimedia information state. Apart from the normal flow illustrated in the figure below, two additional functionalities are supported: the return to previous state and the restart, which changes back to the idle state.

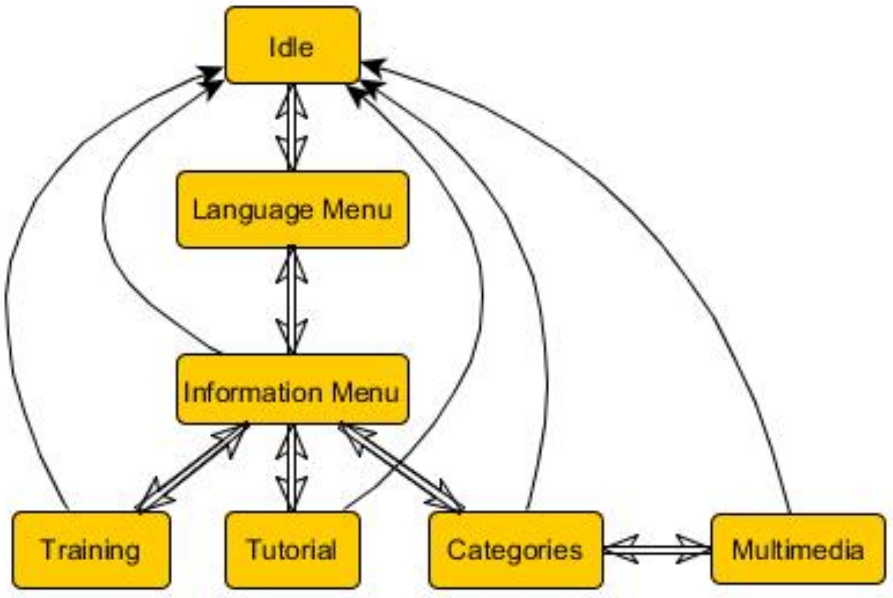


Diagram 9 : Global State Machine

Upon change from one state to another, all the components needed are created, added, modified or deleted by retrieving data from the data collections and through functionality scripts. All the components, effects and scripts needed in every state are created at runtime and deleted when the state changes to the previous one and hidden when a state goes to the next one, according to the regular flow of the state machine. In the following sections the state machine will be described thoroughly.

- Idle State

This state is when no interaction is performed with the system and no one is using it.

- Language Menu State

When the user reaches the system the environment senses his/her presence and informs the system through events. A Language grid appears to the user using interpolation

effect along with an informative label which instructs the user to hold his/her hand over an element, waiting for the user's selection. The virtual assistant shows the language options through an animation. An interactive button-style return to previous state item is also displayed. When the user rise his/her hand the virtual hand cursor appears and moves according to the user's hand movements. During user's interaction with the flag items or the return button, animations are performed through the animator controllers and the callback functions from the intersection with the virtual cursor. Upon flag selection, the language and the state is updated, language menu is hidden and replaced by the information menu.

- Information Menu State

When the information menu is in position, the virtual character informs users that they can select any of the three main information sub-systems or return to the previous state. Additionally, an interactive button is displayed which refers to the sound of the system, a mute button. The user can select it in order to mute environment sounds and the virtual human. Subtitles are automatically displayed so as to inform textually the user. If the user leaves the system returns to the idle state, removing all the components displayed from the scene. Components are removed also when the user chooses to go to previous state. Finally, the user is able to select one of the information elements displayed. Upon selection, the menu disappears and the displayed environment changes through camera movement. The camera is transformed so as to display all the needed components and the assistant takes the appropriate position at the scene.

- Tutorial State

When in Tutorial state (Fig. 64), the assistant is positioned in the left side of the screen through rotation and interpolation, while the projection screen is placed in the middle. A menu is created and placed at the right side of the display, displaying the chapters and the required manipulators (i.e. buttons). A controller container is also created and added under the projection screen for providing the user an information manipulator for navigation and changing the view through functionality scripts as described in chapter [7.3.2]. The information that a user can explore may be images, videos or text. The assistant provides additional descriptive information in the selected language through animations and speech, according to the loaded tutorial data. In order to change state, the user selects the return button to go to previous state and automatically the scene changes and returns to

the view of the information menu state, removing all the unnecessary components from the tutorial state and repositioning the camera and the virtual human through the interpolation and rotation effect.

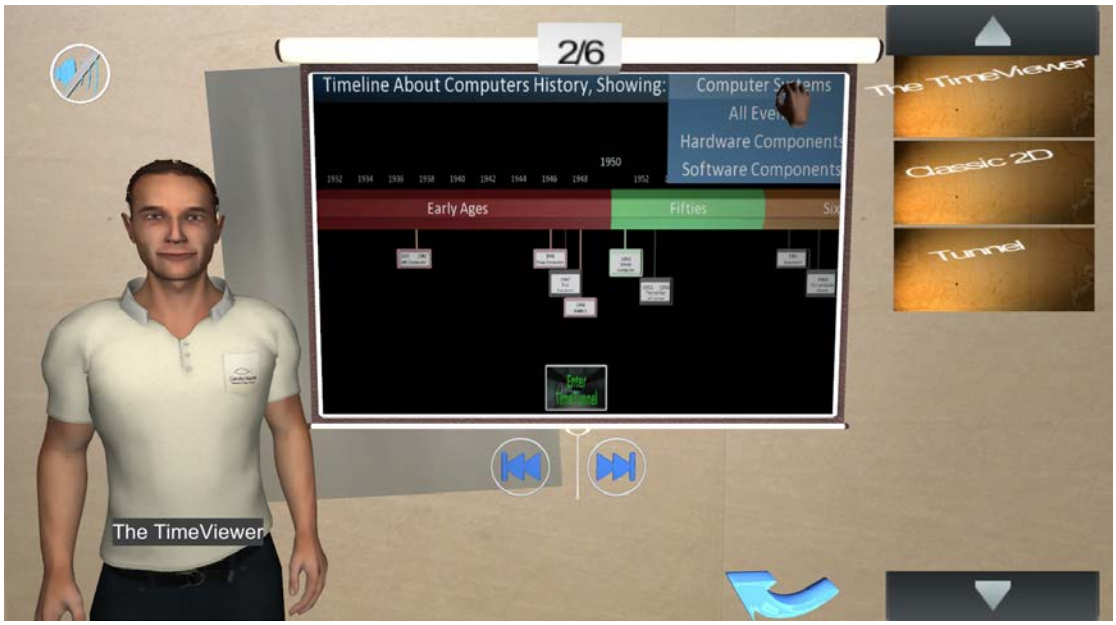


Fig. 64 : Tutorial State View

- Training State

In this state the assistant is located in the center of the display and his full body is visible in order for the user to have a full view of the animations performed. An interactive list with all the available animations is created and placed on the right side, offering navigation and selection functionality through verbal and non verbal interaction with the user (Fig. 65). The virtual human speaks and shows the selected interaction technique through speech and animations derived from the training data collection. The ability to learn the described interaction techniques is given to users by allowing them perform the animations showed by the assistant. The assistant reacts to successful accomplishment or to failure of the user performing the interaction techniques. The user can at any time return to the previous view by selecting the return button. Upon selection, a process similar to the tutorial change state is presented through repositioning the camera and the virtual human and by removing components that are no more needed.

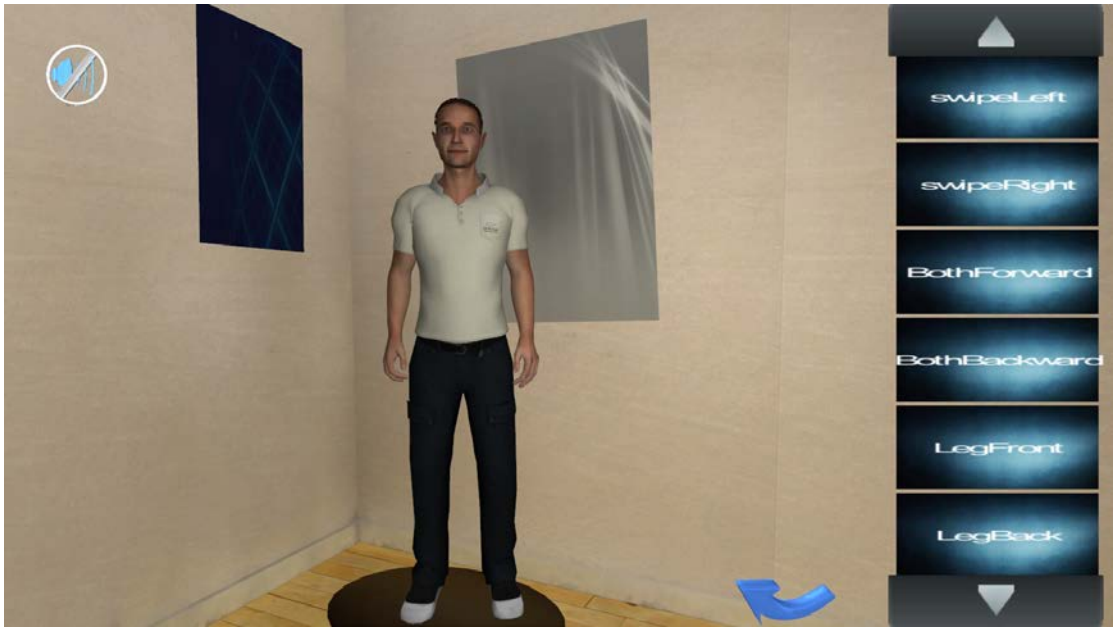


Fig. 65 : Training State View

- Categories State

When the categories (Fig. 72) are selected by the user, a grid of category components appears and the information menu is hidden. The categories are placed in a grid shape containing the data loaded by the XML containing the grid's information, titled accordingly. A navigation system is provided to allow the exploration of additional categories, if any. Upon the cursor being placed over a category, its representative item is displayed. On selection of a specific category, the included information types are displayed around the category. Those components are created at runtime and removed when another category is selected or when returning to previous or initial state. The selection of a type of information, such as images, results in the change of the current state to the multimedia state. The virtual human, the camera and the projection screen are rearranged and the categories are hidden. During this transition, the data that refer to the selected type of the specific category and the animations or speech of the virtual human are retrieved through the category data collection.

- Multimedia State

This state aims to provide interactive visualization of each category's information. The visualization is performed through the information display in the projection screen, the virtual human's animations and descriptive text spoken. Additional multimedia information visualization is accomplished by images, videos, text, 3D models, animations and audio (i.e.

sound and speech described in the next section). The manipulation of multimedia information is accomplished by a controller which is informed of events coming from the environment (e.g. gestures) and cursor selection. The controller contains interactive button-style animated models. Images, Videos and Text are displayed in the projection screen and the controller is located below (Fig. 66). 3D models are located in 3D space with the rotation controllers surrounding them accordingly. Furthermore, the audio is played and can be muted at any time. Finally, synthetic speech is accompanied by facial animations and body gestures in order to be as natural as possible.

Fig. 66 : Image State View

Fig. 67 : Full-screen View

7.3.2.8 Information Visualization and Manipulation

The information visualization at various stages of the system can be achieved through image, video, text, audio and 3D models, (Fig. 68, Fig. 69, Fig. 70, Fig. 71, Fig. 73) as well as the virtual human's gestures and speech synthesis. In every state that a type of information is going to be presented, a script is created which is responsible for the display and the manipulation of it in the area that will be viewed. This script is dynamic and independent of the type of information. Upon selection of the multimedia type in categorized information or of a tutorial's chapter, a process is performed in which the script according to the type of visualized information creates the suitable controller for manipulation, places the information object itself (if it is 3D model) or the GameObject that it will be displayed on (e.g. the projection screen), retrieves the information data from the corresponding data collection and the interaction methods in order to respond to environment events and to the virtual cursor.

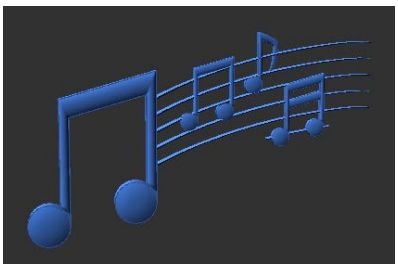


Fig. 68 : Audio Element Visualization

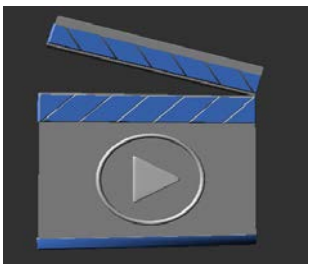


Fig. 69 : Video Element Visualization

Fig. 70 : Image Element Visualization

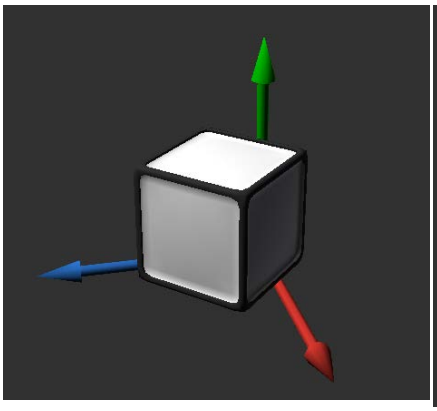


Fig. 71 : 3D model Element Visualization

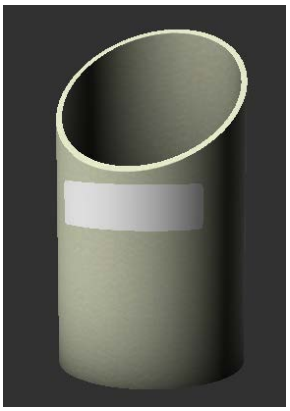


Fig. 72 : Category Element Visualization

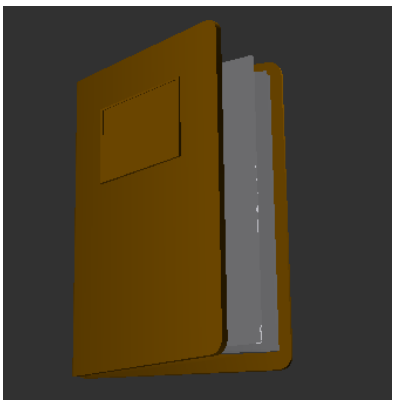


Fig. 73 : Text Element Visualization

The controllers created for each multimedia type contain the following interactive button-style animated elements:

a) Images
    i) Previous
    ii) Next
    iii) Enter/exit full screen display
b) Videos
    i) Previous
    ii) Next
    iii) Enter/exit full screen display
    iv) Play
    v) Pause
c) Text
    i) Previous
    ii) Next
    iii) Enter/exit full screen display
d) Audio
    i) Mute
e) 3D Models
    i) Rotation upwards
    ii) Rotation downwards
    iii) Rotation left
    iv) Rotation right

For all the aforementioned button-style animated elements corresponding methods implement each action in the virtual world. Apart from clicking the buttons using the hand-cursor, the same action is performed when the user provides the corresponding voice command or performs the matching mobile gesture.

8 Evaluation

8.1 Set-up and Participants

The evaluation session took place inside FORTH's Ambient Intelligence Facility. The system was set up in a room with a 55" television display. A total of 10 users participated in the evaluation, 5 females and 5 males. The age of the participants varied from 20 to 35 years old. Six of the users (60%) had intermediate or high computer expertise whereas the other participants had limited expertise. Even though the majority of the users were familiar with computers and touch screen systems, they did not have familiarity with hand gesturing or speech recognition as a mode of interaction with a system.

8.2 The Evaluation Scenario

The scenario of the evaluation involves a virtual human, Bryan, being used as an assistive system to an interactive exhibit in a museum. The interactive exhibit that was assisted was TimeViewer [46, 47], as it is a system with complex interaction which requiring user guidance. Interaction with TimeViewer involves hand and leg gesturing, which users cannot be aware that are offered unless they are being informed accordingly (e.g. users should apply a hand gesture using both of their hands similar to pulling in order to bring an item in front of them). Therefore, assistance and gesture training was imperative for the users to be able to interact with the system, regardless of how natural the gestures were.

8.3 The Evaluation Process

The evaluation process started with the users being informed about the goals of the evaluation process, making clear that the evaluation aimed at assessing the system and not them. Moreover, a consent form that describes the evaluation procedure was given to the users to be signed, stating that any personal information would remain anonymous and strictly confidential.

A series of tasks were assigned to the participants in order to measure the usability of the system. The tasks covered all the primary functionalities of the system in order to assess both the system's design and the interaction process.

User Task List	
Write each task as it will be given to the test participant:	
Task 1:	Choose language and describe what you see.
Task 2:	Go to Tutorial. • Watch the virtual human's presentation.

	• Explain what you see around and then go to the main menu. • What can a user view in time viewer system according to this presentation?
Task 3:	Go to Training • What gestures are used in time viewer? • Did you have any difficulties while doing the gestures? • Did you understand the virtual human's gestures-animations? • If you want to move forward in the tunnel of TimeViewer what are you going to do?
Task 4:	Go to Categories • What do you think this view provides? • Select 2D category. • How can the information be presented; • Choose to see images • Move to the third image. • Enlarge the view and then shrink it.
Task 5:	If you don't want to listen to the virtual human any more what will you do?
Task 6:	Change the language

Furthermore, an additional series of tasks were given for the usage of TimeViewer so as to measure the effectiveness of the assistance provided by the virtual human. During the evaluation process, the evaluators provided assistance only when asked for, in order to examine whether users were able to manipulate the system and utterly depict the effectiveness of the virtual assistant.

User Task List	
Write each task as it will be given to the test participant:	
Task 1:	Navigate for a while into the Time Viewer System
Task 2:	Go to the 3D view.
Task 3:	Use your hands only to navigate into the tunnel.
Task 4:	Now use your legs only to navigate into the tunnel and when needed use your hands to select items.

In addition to the tasks, the participants filled in questionnaires in order to assess the opinion of the users and retrieve qualitative results in a formal way. Furthermore, the participants were encouraged to express their thoughts throughout the evaluation process, which were written down using notes. Finally, a system usability scale (SUS - REFERENCE) questionnaire was assigned in order measure the usability of the system in terms of design, interaction and effectiveness as an assistive tool.

The evaluation process was designed to cover the needs of this thesis; however, more extensive assessment could take place in order to further enrich the outcomes of this evaluation. Firstly, the number of participants could be increased and cover a broader age range. Additionally, it would be very interesting to conduct the evaluation process in a museum containing interactive exhibits and assess different modalities of the framework, even in comparison.

8.4 Results

8.4.1 System Design

The users found the virtual human very helpful and pleasurable to interact with. The information displayed by the system was clear and users did not face difficulties in perceiving any aspect of the system. Although there were a few comments regarding some design decisions, the users were overall found the user interface self-explaining and intuitive.

A common remark was that the tutorial's title was not evident for 80% of the users. The users proposed the increase of the title's size and its placement at a different location, e.g. above the projection screen or next to the mute button. Furthermore, 20% of the users (2 out of 10) could not figure what the chapters in the tutorial stood for and had the impression that their selection would lead them to another view.

Although users understood that the representation of a 3D item had to do with three dimensions, 70% if the users were unable to perceive that the element was meant to present information by 3D models.

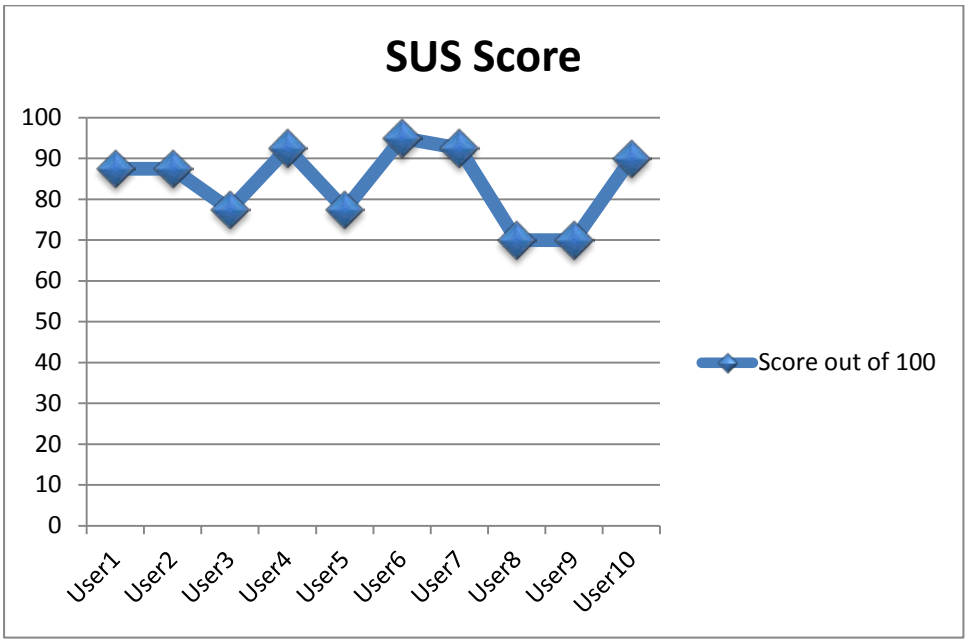
8.4.2 System Usability Scale (SUS) Score

The System Usability Scale (SUS) provides a reliable tool for measuring usability. It consists of a 10 item questionnaire with five response options for the participants; from strongly agree to strongly disagree. It allows the evaluation of a wide variety of products and services. SUS has become an industry standard. The benefits of using SUS include that it:

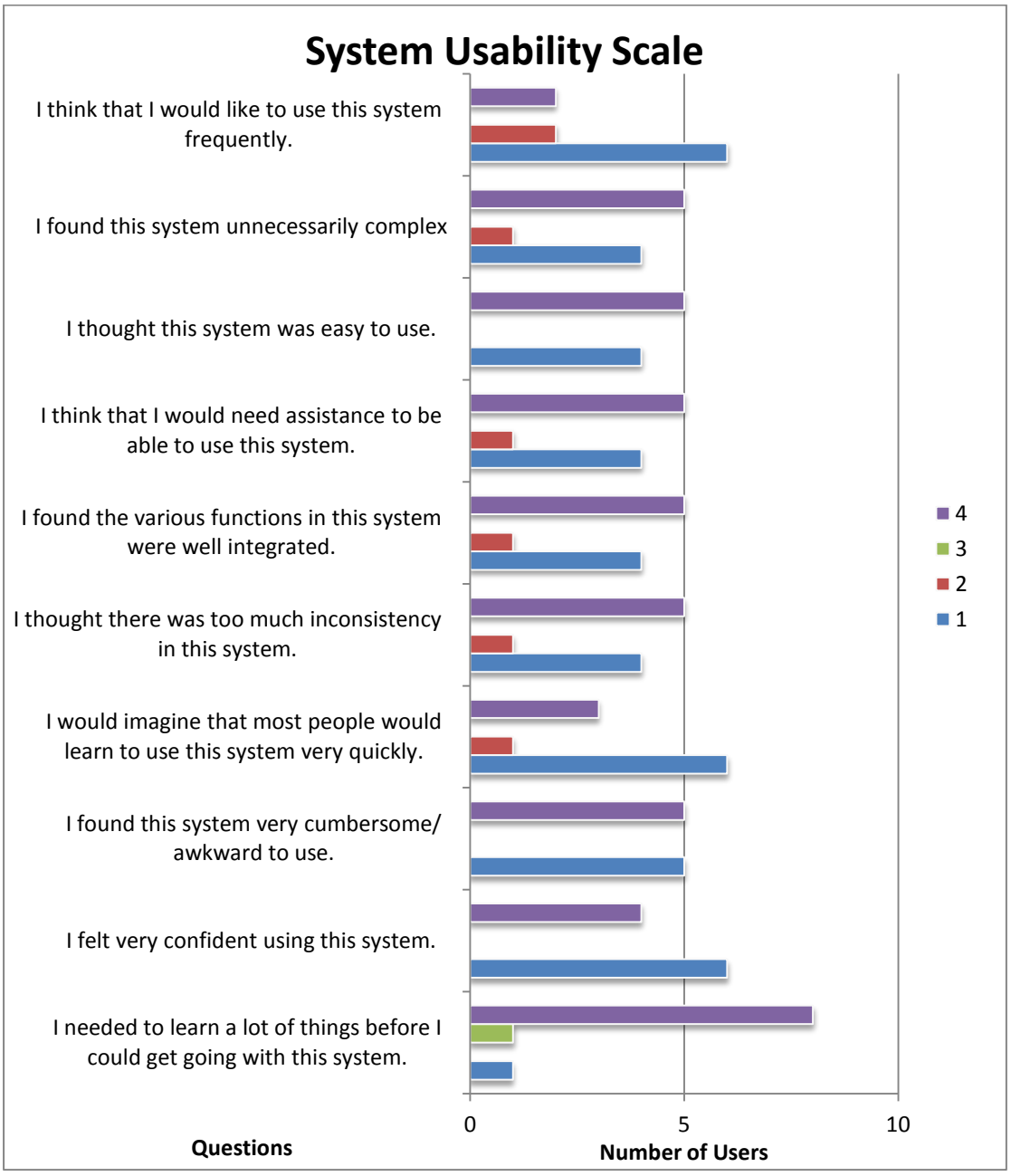
- Is a very easy scale to administer to participants
- Can be used on small sample sizes with reliable results
- Is valid – it can effectively differentiate between usable and unusable systems

The table below contains the questions answered by the participants.

SUS Questionnaire
1. I think that I would like to use this system frequently.
2. I found this system unnecessarily complex.
3. I thought this system was easy to use.
4. I think that I would need assistance to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found this system very cumbersome/awkward to use.
9. I felt very confident using this system.
10. I needed to learn a lot of things before I could get going with this system.



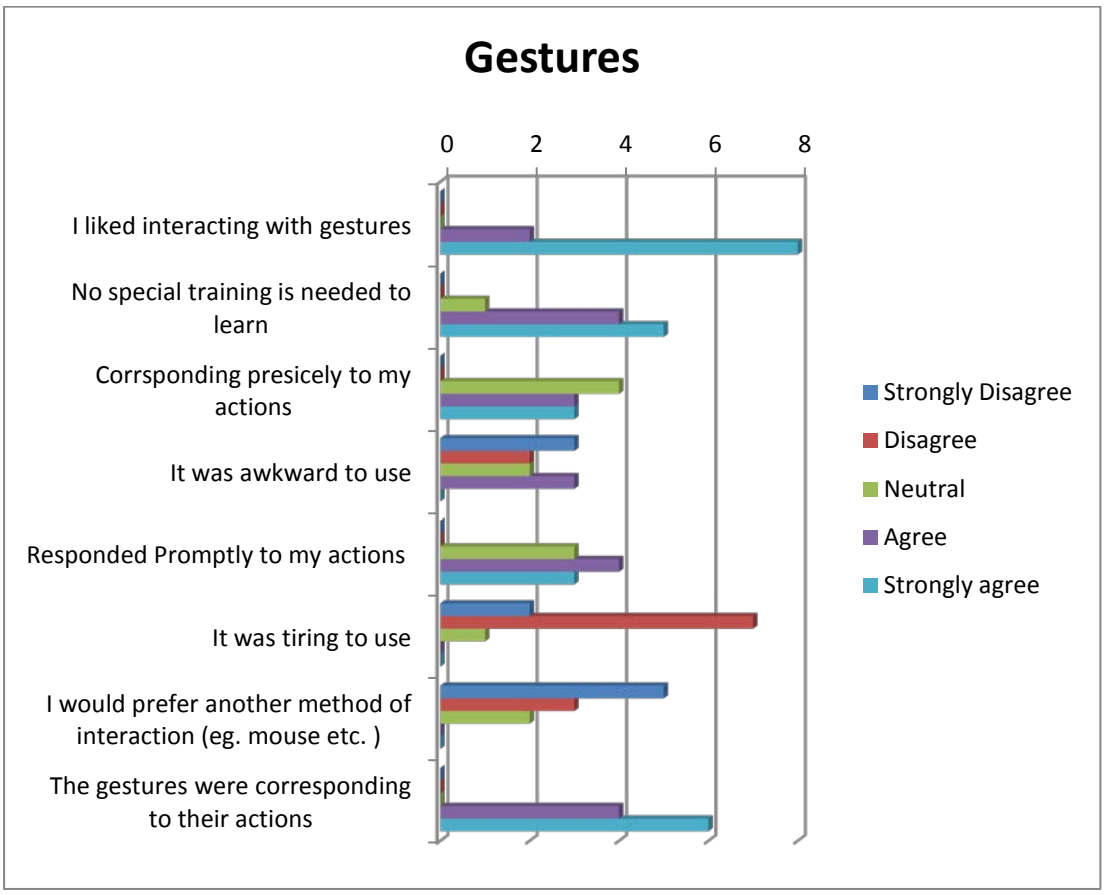
A SUS score above a 68 would be considered above average and anything below 68 is below average. The results of System Usability Scale were very encouraging with a final score of 84. The graph below displays in detail the exact results based on the participant's filled questionnaires and the graph above displays the SUS score per participant (ranging from 70 to 92.5).



8.4.3 Gesture Interaction Results

Gestural interaction involved hand movement in any direction and the manipulation of a virtual cursor. The users performed the supported gestures without difficulty and found them intuitive for specific tasks, such as displaying the next image in a slideshow.

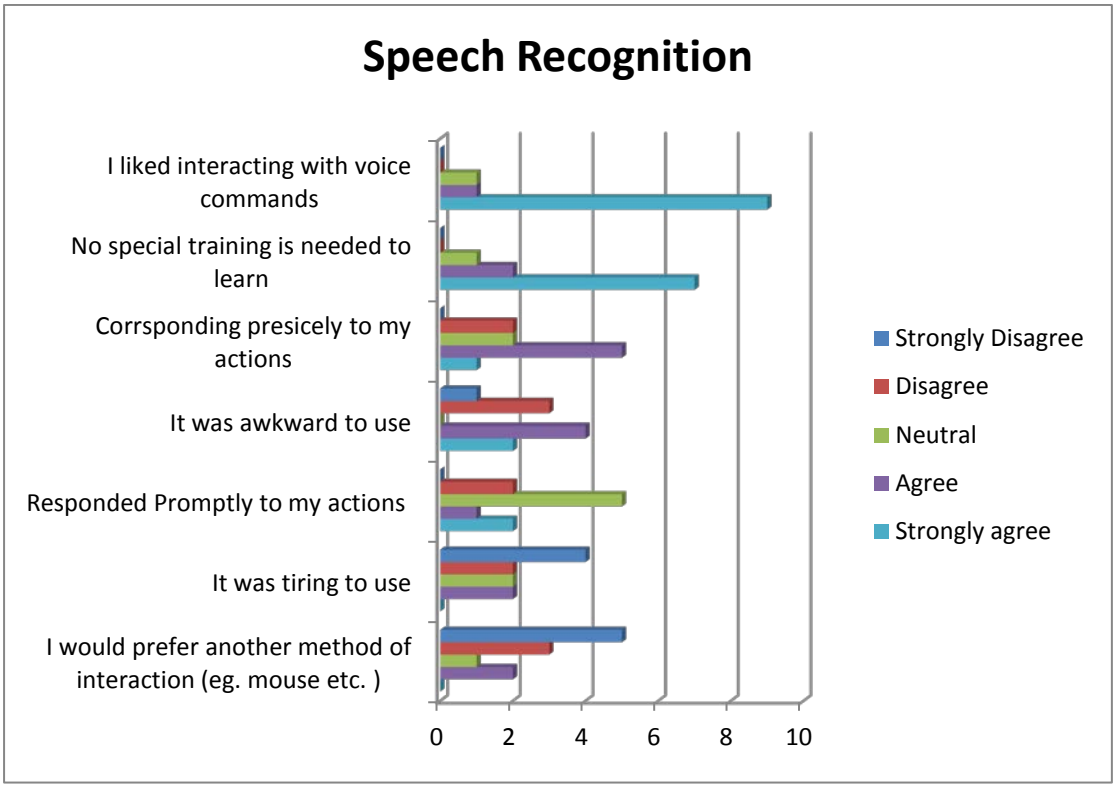
Furthermore, they were able to control the virtual cursor in order to select items shown at the display. Four of the users (40%) initially faced difficulties using the cursor when moving it to the lower part of the display, as tracking was sometimes interrupted. However, once they got familiar to it they could manipulate the system without problem. An additional observation concerning the virtual cursor was that the users found unnecessary that the virtual hand pointed when placed over a selectable item.



8.4.4 Speech Recognition Results

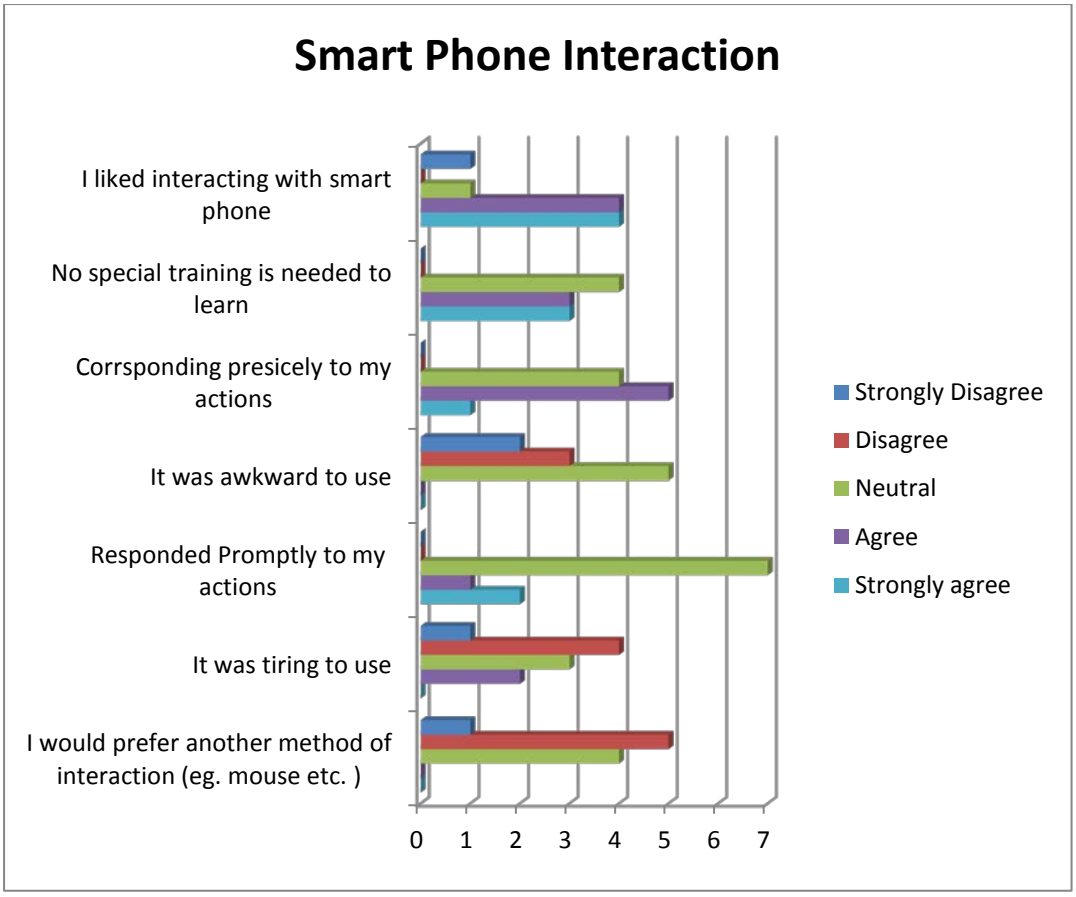
The participants were provided with the list of all the available voice commands in order to assess the responsiveness, the exactness and the usefulness of speech recognition. Despite not being accurate at all times, users generally (80%) managed to manipulate the system using voice commands with a maximum of three tries. Despite that the system did

not focus on human-assistant conversation, voice recognition proved to be pleasing and successful, motivating future research in this area. The graph below displays the users' answers to the questionnaire.



8.4.5 Smart Phone Interaction Results

A smart phone was given to the participants of the evaluation and the available gestures were described. The users enjoyed interacting using mobile devices but preferred the other interaction techniques (hand gestures and speech recognition) which were considered more natural. The graph below illustrates the answers provided to the questionnaires regarding smart phone interaction.



8.4.6 Effectiveness in manipulating TimeViewer

The users were able to start interacting with TimeViewer immediately after they were informed by Bryan. They were aware of the supported gestures and knew what to expect to see in the system. Especially after training, users were confident and able to perform the gestures right ahead.

Indicatively, 3 of the users (30%) stated without being asked that the training was very helpful as they considered themselves experienced users after experimenting using the virtual assistant as they had “tried it in action before”. The confidence was so apparent that one of the users stated his belief that “gesture tracking is more precise in TimeViewer”, although the same software and hardware was used in both cases.

Supplementary to the assigned tasks, the users were asked the following questions in order to extract additional information and urge them to express their opinion regarding the efficiency and the helpfulness of the virtual assistant.

- Did Bryan assist you to manipulate with the Time Viewer?
- What else would you like to be offered by the Bryan in order to help you?

- What would you like not to be presented by Bryan, which you believe was unnecessary?
- What didn't help you at all?
- What haven't you understood in the system?

The users believed that TimeViewer would be difficult to use without interactive guidance, as leaflets for instance would not be sufficient. Moreover, Bryan was described as sufficient, comprehensive and did not display redundant information. The users were familiar with all the aspects of TimeViewer and only 2 of them (20%) did not recognize one functionality (events' categorization) as its description was considered not extensive enough in the assistive system.

9 Conclusions and Future Work

This thesis reports on the design, the development and the evaluation of a framework which implements virtual humans behaviors by creating body gestures and speech synthesis that can be used for information provision, creates interactive multimedia information visualizations (e.g. images, text, audio, videos, 3D models) and implements communication through multimodal interaction techniques. The presented framework provides a dynamic data modeling mechanism for storage and retrieval.

The supported communication capabilities include human to agent, agent to environment or agent to agent interaction. The generated virtual agents can have diverse roles, as they can be used as assistive tools for existing systems, standalone "applications" or even as vital parts of smart environments.

When acting as an assistant, the framework provides the tools needed for presenting information in the form of tutorials and real-time help. Furthermore, the virtual humans support the users' training on interaction techniques applied for the assisted system. As standalone application, the framework provides categorized information visualization and interactive help about the interaction techniques and information presented. Finally, when the virtual humans are embedded in other environments the framework is be able to create hybrid mode that supports all the aforementioned functionalities individually or even in combination.

The developed framework supports multiple multimodal techniques in order to fit to various ambient intelligence environments and offer natural interaction, such as gestural, verbal and tangible interaction, in a wide range of setups.

An evaluation study with ten participants was conducted in order to assess the framework in terms of usability, effectiveness and likeability. The results of the evaluation were very promising and boost further research in the domain of interactive virtual humans in ambient intelligence environments.

Based on the encouraging results of the evaluation process, further research is planned in the domain of virtual humans in ambient intelligence environments. Improvements in the system's design are also planned based on the users' comments. Interaction is planned to be enhanced with further gestural techniques, as users were amused by kinesthetic interaction. Furthermore, the virtual humans' training capabilities can be used in the context of a virtual assistant as a means of measuring the interaction techniques success rates and producing statistical reports depicting their usability.

Speech recognition proved to be very popular among the users and the conversational abilities of the virtual humans are intended to be improved. Moreover, it would be interesting to experiment with additional virtual humans with distinct roles in the virtual world for collaboration in complex environments.

A drawback that was evident when deploying the framework is the provision of content input to the system, which can be addressed by the development of a tool for content insertion and editing. Finally, since the framework is built on top of an engine that supports interoperability with mobile devices; further work can be done on adapting the virtual humans to suit the needs of mobility and small displays.

10 References

1. Weiser M. (1991). The Computer for the Twenty-First Century
2. Stephanidis C., Antona M. (2013). Universal Access in Human-Computer-Interaction. User and Context Diversity.
3. Murray, J. (2013). "Composing Multimodality".
4. Weber w., Rabaey J. and Aarts E. (2005). Ambient Intelligence
5. Augusto Juan Carlos, Nakashima Hideyuki , Aghajan Hamid (2010), Handbook of Ambient Intelligence and Smart Environments
6. Georgalis Yannis, Grammenos Dimitris , Stephanidis Constantine (2009). Middleware for Ambient Intelligence Environments : Reviewing Requirements and Communication
7. Cook Diane J., Augusto Juan C., Jakkula R. Vikramaditya (2009).Ambient intelligence: Technologies, applications, and opportunities
8. Turk Matthew (2014). Multimodal interaction: A review
9. Microsoft Kinect http://www.xbox.com/kinect/
10. Grzejszczak Tomasz, Mikulski Michał, Szkodny Tadeusz, Jędrasiak Karol (2013). Gesture Based Robot Control
11. Grammenos D., Zabulis X., Michel D., Sarmis Th., Tzavanidis K., Argyros A. and Stephanidis C. (2011). Design and Development of Four Prototype Interactive Edutainment Exhibits for Museums
12. Suarez Jesus, Murphy Robin R. (2012). Hand gesture recognition with depth images: A review
13. Biswas K. K., Basu Saurav Kumar (2011). Gesture recognition using Microsoft Kinect
14. Ren Zhou, Meng Jingjing, Yuan Junsong , Zhang Zhengyou (2011). Robust hand gesture recognition with kinect sensor
15. Povey Daniel ,Ghoshal Arnab ,Boulianne Gilles, Burget Lukas, Glembek Ondrej, Goel Nagendra , Hannemann Mirko, Motlicek Petr, Qian Yanmin, Schwarz Petr, Silovsky Jan, Stemmer George, Vesely Karel (2011). The Kaldi Speech Recognition Toolkit
16. Kumar Kuldeep (2011). Hindi Speech Recognition System Using HTK
17. Kurtenbach,G., Hulteen (1990). Gestures in Human-Computer Communication
18. Saffer Dan (2008).Designing Gestural Interfaces: Touchscreens and Interactive Devices
19. Meng Rufeng, Isenhower Jason, Qin Chuan, Nelakuditi Srihari (2012). Can Smartphone Sensors Enhance Kinect Experience ?

20. Ruiz Jaime, Li Yang, Lank Edward (2011). User-defined motion gestures for mobile interaction
21. Kray Christian, Nesbitt Daniel, Dawson John, Rohs Michael (2010). User-Defined Gestures for Connecting Mobile Phones , Public Displays , and Tabletops
22. Russell, S. and Norvig, P. (1995). Artificial Intelligence: A Modern Approach
23. N. Kasabov,(1998). Introduction: Hybrid intelligent adaptive systems
24. Hartholt Arno, Traum David, Marsella Stacy, Shapiro Ari, Stratou Giota, Morency Louis-philippe, Gratch Jonathan (2013). All Together Now, Introducing the Virtual Human Toolkit
25. Yaghoubzadeh Ramin, Kramer Marcel, Pitsch Karola, Kopp Stefan (2013). Virtual agents as daily assistants for elderly or cognitively impaired people Studies on acceptance and interaction feasibility
26. Feng Andrew, Huang Yazhou, Xu Yuyu, Shapiro Ari (2012). Automating the Transfer of a Generic Set of Behaviors Onto a Virtual Character
27. Gratch Jonathan, Morency Louis-philippe, Scherer Stefan, Stratou Giota (2012). User-State Sensing for Virtual Health Agents and TeleHealth Applications
28. Karouzaki Effie, Savidis Anthony (2012). A Framework for Adaptive Game Presenters with Emotions and Social Comments
29. Lane H Chad, Noren Dan, Auerbach Daniel, Birch Mike, Swartout William (2011). Intelligent Tutoring Goes to the Museum in the Big City : A Pedagogical Agent for Informal Science Education
30. Swartout William, Traum David, Artstein Ron, Noren Dan, Debevec Paul, Bronnenkant Kerry, Williams Josh, Leuski Anton, Narayanan Shrikanth, Piepol Diane, Lane Chad, Morie Jacquelyn, Aggarwal Priti, Liewer Matt, Chiang Jen-yuan, Gerten Jillian, Chu Selina, White Kyle (2010). Ada and Grace : Toward Realistic and Engaging Virtual Museum Guides
31. Zhang Hui, Fricker Damian, Smith Thomas G, Yu Chen (2010). Real-Time Adaptive Behaviors in Multimodal Human- Avatar Interactions
32. Leuski Anton, Traum David, Rey Marina (2010). NPCEditor : A Tool for Building Question-Answering Characters
33. Meschtscherjakov Alexander, Reitberger Wolfgang, Mirlacher Thomas, Huber Hermann, Tscheligi Manfred (2009). AmIQuin - An Ambient Mannequin for the Shopping Environment
34. Baldassarri Sandra, Cerezo Eva, Seron Francisco J. (2008). Maxine: A platform for embodied animated agents

35. Cassell Justin, Vilhjálmsson Hannes Högni, Bickmore Timothy ,(2001). BEAT: the Behavior Expression Animation Toolkit
36. Santangelo Antonella, Augello Agnese, Sorce Salvatore, Pilato Giovanni, Gentile Antonio, Genco Alessandro, Gaglio Salvatore (2007). A Virtual Shopper Customer Assistant in Pervasive Environments
37. Berhe, G., Brunie, L., Pierson, J.M. (2004). Modeling Service-Based Multimedia Content Adaptation.
38. Cassell, Justine; Prevost, Scott; Sullivan, Joseph; Churchill, Elizabeth (2000). Embodied Conversational Agents, Cambridge
39. Kunc Ladislav, Kleindienst Jan (2007). ECAF : Authoring Language for Embodied Conversational Agents
40. Petit-Rozé Christelle, Grislin-Le Strugeon Emmanuelle (2006). MAPIS, a multi-agent system for information personalization
41. Ndiaye Alassane, Gebhard Patrick, Kipp Michael, Klesen Martin, Schneider Michael, Wahlster Wolfgang (2005). Ambient Intelligence in Edutainment : Tangible Interaction with Life-Like Exhibit Guides
42. Cavalluzzi Addolorata, Carolis Berardina De, Pizzutilo Sebastiano, Cozzolongo Giovanni (2004). Interacting with Embodied Agents in Public Environments
43. Chaudhuri, Parag, George Papagiannakis, and Nadia Magnenat-Thalmann (2008). Self adaptive animation based on user perspective.
44. Magnenat-Thalmann, Nadia, George Papagiannakis, and Parag Chaudhuri (2008). Applications of Interactive Virtual Humans in Mobile Augmented Reality.
45. Papagiannakis, George (2013). Geometric algebra rotors for skinned character animation blending.]
46. Drossis, Giannis, Dimitris Grammenos, Ilia Adami, and Constantine Stephanidis (2013). 3D Visualization and Multimodal Interaction with Temporal Information Using Timelines.
47. Drossis, Giannis, Dimitris Grammenos, Maria Bouhli, Ilia Adami, and Constantine Stephanidis (2013). Comparative evaluation among diverse interaction techniques in three dimensional environments
48. Abbattista F., Catucci G., Semeraro G, Zambetta F. (2004). SAMIR : A Smart 3D Assistant on the Web
49. Schiaffino Silvia, Amandi Analía (2004). User – interface agent interaction: personalization issues
50. Rickel Jeff, Johnson W.Lewis (2002). Extending Virtual Humans to support Team Training in Virtual Reality

51. Balcisoy Selim, Kallmann Marcelo, Torre Rémy, Fua Pascal, Thalmann Daniel (2002). Interaction Techniques with Virtual Humans in Mixed Environments
52. Traum David, Rickel Jeff (2002). Embodied Agents for Multi-party Dialogue in Immersive Virtual Worlds
53. Tsotros Manolis (2002). Archeoguide : An Augmented Reality Guide for Archaological Sites
54. Maher Mary Lou (2002). Agent Models of 3D Virtual Worlds
55. Lee Jehee, Chai Jinxiang, Reitsma Paul S. A., Hodgins Jessica K., Pollard Nancy S. (2002). Interactive Control of Avatars Animated with Human Motion Data
56. Cassell Justine (2001). Conversational Agents Representation and Intelligence in User Interfaces
57. Thalmann Daniel (2001). The Role of Virtual Humans in Virtual Environment Technology and Interfaces
58. http://www.quora.com/What-is-the-definition-of-personalization , last accessed on 20/3/2014
59. Bell Mark, (2008). Virtual Worlds Research: Past, Present & Future
60. Nijholt Anton, Hulstijn Joris (2000). Multimodal Interactions with Agents in A Virtual Worlds
61. Christian Andrew D, Avery Brian L, Christian Andrew , Avery Brian (2000). Speak Out and Annoy Someone : Experiences with Intelligent Kiosks
62. Evers Marc, Nijholt Anton (2000). Jacob - An animated instruction agent in virtual reality
63. Grammenos, D., Margetis, G., Koutlemanis, P., Zabulis, X. (2012). Paximadaki, the game: Creating an advergame for promoting traditional food products.
64. Jung Yvonne., Kuijper Arjan, Fellner Dieter, Kipp Michael, Miksatko Jan, Gratch Jonathan, Thalmann Daniel (2011). Believable Virtual Characters in Human-Computer Dialogs.
65. Kasap Zerrin, Magnenat-Thalmann Nadia (2007). Intelligent virtual humans with autonomy and personality: State-of-the-art.
66. Gillies Marco, Spanlang Bernhard (2010). Comparing and Evaluating Real Time Character Engines for Virtual Environments.
67. Egges Arjan, Papagiannakis George, Magnenat-Thalmann Nadia (2007). Presence and interaction in mixed reality environments.
68. Jack Rachael, Caldara Roberto, Schyns Philippe (2011). Internal Representations Reveal Diversity in Expectations of Facial Expressions of Emotion.

69. Drossis Giannis, Grammenos Dimitris, Birliraki Chryssi, Stephanidis Constantine (2013). MAGIC: Developing a Multimedia Gallery Supporting mid-Air Gesture-Based Interaction and Control.
70. Kenny Patrick, Hartholt Arno, Gratch Jonathan, Swartout William, Traum David, Marsella Stacy, Piepol Diane (2007). Building Interactive Virtual Humans for Training Environments.
71. Leonidis, A., Korozi, M., Margetis, G., Grammenos, D., & Stephanidis, C (2013). An Intelligent Hotel Room
72. MacMillan Dictionary Online, http://www.macmillandictionary.com/dictionary/british/ambient
73. Oxford Dictionaries Online, http://oxforddictionaries.com/definition/english/ambient
74. Hidden Markov Model Toolkit (HTK) speech Recognition, http://htk.eng.cam.ac.uk/develop/atk.shtml
75. Virtual Human Toolkit, https://vhtoolkit.ict.usc.edu
76. Specification for a Standard Humanoid (H-Anim), http://h-anim.org/
77. OSG: OpenSceneGraph 2.0 (2007). http:// www. openscenegraph.org
78. http://en.wikipedia.org/wiki/
79. OpenNI, http://www.openni.org/
80. Unity3d, http://unity3d.com/ and documentation http://docs.unity3d.com/Documentation
81. Serialization, http://msdn.microsoft.com/en-us/library/7ay27kt9%28v=vs.110%29.aspx