Computer Science Department
Faculty of Sciences and Engineering
University of Crete

Supporting Young Children in Ambient Intelligence Environments

Emmanouil Zidianakis

Ph.D. Dissertation

March 2015

COMPUTER SCIENCE DEPARTMENT
FACULTY OF SCIENCES AND ENGINEERING
UNIVERSITY OF CRETE

Supporting Young Children in Ambient Intelligence Environments

EMMANOUIL ZIDIANAKIS

PH.D. DISSERTATION

MARCH 2015

Supporting Young Children in Ambient Intelligence Environments

Emmanouil Zidianakis

MARCH 2015

COMPUTER SCIENCE DEPARTMENT

FACULTY OF SCIENCES AND ENGINEERING

UNIVERSITY OF CRETE

Thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy[1].

Doctoral Thesis Committee:	Constantine Stephanidis, Professor, University of Crete (Advisor)
	Anthony Savidis, Professor, University of Crete
	Xenophon Zabulis, Principal Researcher, ICS-FORTH. Visiting Professor, University of Crete
	George Papagiannakis, Assistant Professor, University of Crete
	Panayiotis Zaphiris, Professor, Dean of School of Fine and Applied Arts, Department of multimedia and graphic arts, Cyprus university of technology
	Dimitris Grammenos, Principal Researcher, ICS-FORTH
	Margherita Antona, Principal Researcher, ICS-FORTH

[1] The work reported in this thesis has been conducted at the Human Computer Interaction (HCI) Laboratory of the Institute of Computer Science (ICS) of the Foundation for Research and Technology – Hellas (FORTH), and has been financially supported by a FORTH-ICS scholarship. Additionally, this work has been supported by the FORTH-ICS internal RTD program "Ambient Intelligence Environments".

Computer Science Department

Faculty of Sciences and Engineering

University of Crete

Supporting Young Children in Ambient Intelligence Environments

Dissertation submitted by Emmanouil Zidianakis. In partial fulfillment of the requirements for the Doctor of Philosophy degree in Computer Science.

Author

Εμμανουήλ Ζηδιανάκης
Emmanouil Zidianakis

Examination Committee

Καθηγητής Κωνσταντίνος Στεφανίδης (Επόπτης), Πανεπιστήμιο Κρήτης
Professor Constantine Stephanidis (Advisor), University of Crete

Καθηγητής Αντώνης Σαββίδης, Πανεπιστήμιο Κρήτης
Professor Antonis Savidis, University of Crete

Κύριος Ερευνητής Δρ. Ξενοφών Ζαμπούλης, Ινστιτούτο Πληροφορικής ΙΤΕ
Principal Researcher Dr. Xenophon Zaboulis, ICS-FORTH

Examination Committee (cont.)

Επίκουρος Καθηγητής Γεώργιος Παπαγιαννάκης, Πανεπιστήμιο Κρήτης
Assistant Professor Georgios Papagianakis, University of Crete

Κύριος Ερευνητής Δρ. Δημήτρης Γραμμένος, Ινστιτούτο Πληροφορικής ΙΤΕ
Principal Researcher Dr. Dimitris Grammenos, ICS-FORTH

Κύρια Ερευνήτρια Δρ. Μαργκερίτα Αντόνα, Ινστιτούτο Πληροφορικής ΙΤΕ
Principal Researcher Dr. Margherita Antona, ICS-FORTH

Καθηγητής Παναγιώτης Ζαφείρης, Τεχνολογικό Πανεπιστήμιο Κύπρου
Professor Panayiotis Zaphiris, Cyprus University of Technology

Department Approval

Παναγιώτης Τσακαλίδης, Πρόεδρος τμήματος Επιστήμης υπολογιστών
Panagiotis Tsakalides, Professor, Chairman of the Computer Science Department

Heraklion, March 2015

This thesis is dedicated to my inspiring parents Καλλιόπη, Γιώργος and my sister Ιωάννα, who have supported me all the way since the beginning of my studies and never stop giving of themselves in countless ways.

Also, this thesis is dedicated to my friends who have been a great source of motivation and encouragement.

Finally, this thesis is dedicated to all those who believe in the richness of learning...

Declaration of Originality

I declare that this thesis is my own work. Information derived from published or unpublished work of others has been formally acknowledged.

Acknowledgements

I owe many thanks to my thesis supervisor, Professor Constantine Stephanidis for his strong support and guidance through my Ph.D. studies. Professor Stephanidis, Professor Savidis and Dr. Xenophon Zabulis, helped me to improve my research skills with their close attention to my work and critical thinking. The ideas, guidance and support of Dr. Margherita Antona were invaluable in order to shape this work to its current state.

I am also grateful to my friends and colleagues Nikolaos Partarakis, Danae Ioannidi, Giannis Drossis, Nikolas Kazepis and for their support. I would like also to thank my beloved friend Eirini Kontaki who had a critical contribution providing me access to the literature required for this work.

I would also like to thank Kallia Stratigi, Ioanna Zidianaki, Antonis Ntelidakis and Antonis Hatjiantoniou for their invaluable suggestions and comments, which also provided clear guidance towards exploring interesting future directions of my work.

Finally, I would like to thank all my friends and colleagues at the Institute of Computer Science of the Foundation for Research and Technology, Hellas (ICS-FORTH), who supported me and my work and helped in making the office environment stimulating, productive and pleasant.

ΥΠΟΣΤΗΡΙΞΗ ΜΙΚΡΩΝ ΠΑΙΔΙΩΝ ΣΕ ΣΥΓΧΡΟΝΑ ΠΕΡΙΒΑΛΛΟΝΤΑ ΔΙΑΧΥΤΗΣ ΝΟΗΜΟΣΥΝΗΣ

Εμμανουήλ Ζηδιανάκης

Διδακτορική Διατριβή

Τμημα Επιστημης Υπολογιστων

Σχολη Θετικων και Τεχνολογικων Επιστημων

Πανεπιστημιο Κρητης

Περίληψη

Η Διάχυτη Νοημοσύνη (Ambient Intelligence - AmI) αποτελεί το όραμα της κοινωνίας της πληροφορίας του μέλλοντος που απορρέει από την σύγκλιση πανταχού παρόντων υπολογιστικών συστημάτων, της επικοινωνίας και των ευφυών διεπαφών. Πολλές εφαρμογές της Διάχυτης Νοημοσύνης έχουν ως στόχο τη βελτίωση και την υποστήριξη της καθημερινότητας ενός συνόλου ομάδων χρηστών, το οποίο περιλαμβάνει και μη παραδοσιακούς χρήστες διαδραστικών τεχνολογιών όπως είναι τα μικρά παιδιά. Ωστόσο, τα πλεονεκτήματα που προκύπτουν από την εφαρμογή της Διάχυτης Νοημοσύνης για τα μικρά παιδιά δεν έχουν ακόμη εξερευνηθεί σε ικανοποιητικό βαθμό. Η παρούσα διατριβή έχει ως στόχο την δημιουργία ενός περιβάλλοντος Διάχυτης Νοημοσύνης ικανό να υποστηρίξει την ανάπτυξη μικρών παιδιών μέσω του παιχνιδιού. Προκειμένου να επιτευχθεί αυτός ο στόχος, προτείνεται ένα τεχνολογικό πλαίσιο (framework) για να υποστηρίξει το σχεδιασμό, την ανάπτυξη και υλοποίηση καινοτόμων παιχνιδιών. Τα παιχνίδια αυτά έχουν τη δυνατότητα να παρακολουθούν και να αξιολογούν τις ικανότητες των παιδιών, ενώ παράλληλα εμπλουτίζουν την εμπειρία του παιχνιδιού καθώς προσαρμόζονται συνεχώς σε αυτές. Παράλληλα, οι ανάγκες των παιδιών για δραστηριότητα, άσκηση και ευχαρίστηση καλύπτονται με το βέλτιστο δυνατό τρόπο, ενώ παράλληλα παρέχονται δυνατότητες για την ενίσχυση της δημιουργικότητας.

Το προτεινόμενο τεχνολογικό πλαίσιο υποστηρίζει την δημιουργία έξυπνων παιχνιδιών που μοιράζονται μοναδικά χαρακτηριστικά, όπως: α) είναι προσαρμόσιμα στις ικανότητες των παιδιών, β) έχουν την δυνατότητα να δημιουργήσουν κατάλληλους διαύλους επικοινωνίας με τα παιδιά (λεκτικούς ή μη), γ) δίνουν στα παιδιά τον έλεγχο

ώστε με την εξάσκηση να μπορούν να παίξουν αυτόνομα, ενισχύοντας τους με αυτόν τον τρόπο την αίσθηση της αυτοπεποίθησης και την ανεξαρτησίας, δ) αυξάνουν το κίνητρο και ε) μειώνουν τον φόβο της αποτυχίας και του λάθους.

Προκειμένου να δημιουργηθεί το προτεινόμενο τεχνολογικό πλαίσιο, οι ενδιαφερόμενοι χρήστες (δηλαδή μικρά παιδιά, οι γονείς τους καθώς και οι ειδικοί στην ανάπτυξη του παιδιού) αρχικά προσδιορίστηκαν και εν συνεχεία έγινε η συλλογή και ανάλυση των απαιτήσεών τους. Με αυτόν τον τρόπο έγινε ο ορισμός των λειτουργικών και μη-λειτουργικών απαιτήσεων, επιτρέποντας τον καθορισμό σεναρίων αλληλεπίδρασης στο πλαίσιο ενός περιβάλλοντος Διάχυτης Νοημοσύνης. Αναφορικά με την υλοποίηση του τεχνολογικού πλαισίου, δημιουργήθηκε μια αρχιτεκτονική διεργασιών βασισμένη σε υπηρεσίες (service-oriented architecture) που περιλαμβάνει επιμέρους υποσυστήματα. Τα επιμέρους υποσυστήματα σχεδιάστηκαν και δημιουργήθηκαν λαμβάνοντας υπόψη μια γενικευμένη αρχιτεκτονική, ούτως ώστε να υποστηρίζουν τον ορισμό και την εξαγωγή διαχείρισης γνώσης, εναλλακτικές τεχνικές αλληλεπίδρασης στο πλαίσιο της Διάχυτης Νοημοσύνης, προσωποποίηση και προσαρμογή. Επιπλέον, το προτεινόμενο τεχνολογικό πλαίσιο υποστηρίζει την αυτόματη εξαγωγή γνώσης σχετικά με τις ικανότητες και της συνολικής ανάπτυξης των παιδιών βάση της παρακολούθησης της αλληλεπίδρασης, ούτως ώστε να προσφέρει ενδείξεις που αφορούν την ανάπτυξή τους σε σχέση με το επίπεδο ωριμότητας των ικανοτήτων τους. Με αυτόν τον τρόπο, η παρεχόμενη τεχνολογική υποδομή επιτρέπει την αναγνώριση τυχόν προβλημάτων ανάπτυξης τα οποία χρήζουν περαιτέρω διάγνωσης και έρευνας. Για το σκοπό αυτό, στους γονείς παρέχονται γενικές πληροφορίες, με έναν ευχάριστο και πρακτικό τρόπο, σχετικά με την σωματική και διανοητική ανάπτυξη των παιδιών τους, καθώς και ενδείξεις πιθανών ανώριμων ικανοτήτων. Τέλος, εκτενή δεδομένα χρήσης καθώς και το πλήρες ιστορικό της αλληλεπίδρασης παρέχονται στους ειδικούς στην ανάπτυξη των μικρών παιδιών για το καθορισμό του αν και κατά πόσο το παιδί κατακτά τα ορόσημα ανάπτυξης. Παράλληλα, η τεχνολογική υποδομή χρησιμοποιείται με τη βοήθεια εργοθεραπευτών ούτως ώστε να δημιουργηθούν καινοτόμοι τρόποι εφαρμογής πρωτοπόρας τεχνολογίας στην πράξη.

Στο πλαίσιο ενός περιβάλλοντος Διάχυτης Νοημοσύνης, η χρήση του τεχνολογικού πλαισίου στην πράξη επικυρώθηκε με τον σχεδιασμό και την υλοποίηση ενός συνόλου από επαυξημένα αντικείμενα, τα οποία δρουν ως υλικά και ψηφιακά μέσα για να επιτρέψουν την αλληλεπίδραση των παιδιών με το σύστημα. Τα αντικείμενα αυτά περιλαμβάνουν ένα επαυξημένο παιδικό τραπέζι, μια επαυξημένη καρέκλα, ένα ψηφιακό στυλό, ψηφιακά ζάρια, κοκ. Επιπλέον, αρκετά φυσικά αντικείμενα όπως κομμάτια από παζλ, κάρτες αναγνώρισης, παιχνίδια, κοκ., τα οποία δεν περιλαμβάνουν τεχνολογία, εμπλουτίστηκαν και επαυξήθηκαν χρησιμοποιώντας εργαλεία που προφέρει το τεχνολογικό πλαίσιο. Για την παροχή επαυξημένων διόδων επικοινωνίας, διάφορες εφαρμογές δημιουργήθηκαν, όπως ένας τρισδιάστατος εικονικός χαρακτήρας-συμπαίκτης, ψηφιακές αναπαραστάσεις γνωστών παραδοσιακών παιχνιδιών. Οι εφαρμογές αυτές αποτελούν το απαραίτητο ενδιάμεσο λογισμικό το οποίο φιλοξενεί τις νέες διόδους επικοινωνίας που προτείνονται από την παρούσα διατριβή. Για να υποστηριχθούν οι ανάγκες των επιμέρους ομάδων χρήσης (με βάση τις ορισμένες απαιτήσεις τους) τόσο για τους γονείς όσο και για τους ειδικούς στην ανάπτυξη των μικρών παιδιών αναπτύχθηκαν εφαρμογές διαχείρισης των εφαρμογών και των συστημάτων, περιλαμβάνοντας ποικίλες εφαρμογές διαχείρισης περιεχομένου και παρακολούθησης της αλληλεπίδρασης. Τέλος, το τεχνολογικό πλαίσιο αυτό καθαυτό είναι ένα εργαλείο έτοιμο για χρήση από τους προγραμματιστές οι οποίοι επιθυμούν να αναπτύξουν καινοτόμες εφαρμογές που υποστηρίζουν παιγνιώδη αλληλεπίδραση σε περιβάλλοντα Διάχυτης Νοημοσύνης. Με αυτή την έννοια, το τεχνολογικό πλαίσιο παρέχει ένα σύνολο από εργαλεία για την δημιουργία, τον ορισμό αλληλεπίδρασης, παρακολούθησης και ενσωμάτωσης ευφυών επαυξημένων αντικειμένων ούτως ώστε να μειώσουν κατακόρυφα την απαιτούμενη προσπάθεια υλοποίησης.

Το τεχνολογικό πλαίσιο έχει εγκατασταθεί στους χώρους προσομοίωσης του κτιρίου Διάχυτης Νοημοσύνης του ΙΤΕ και αξιολογήθηκε σε δύο φάσεις από παιδιά, γονείς και ειδικούς στην ανάπτυξη των μικρών παιδιών, προκειμένου να αναγνωριστούν πιθανά προβλήματα. Η πρώτη φάση της αξιολόγησης έγινε μετά την υλοποίηση της κύριας τεχνολογικής υποδομής, ώστε να υπάρχει μια πρώτη καταμέτρηση της αποδοχής από τους χρήστες και του εργονομικού σχεδιασμού των αντικειμένων. Συμμετέχοντες στην αξιολόγηση ήταν παιδιά τριών έως έξι ετών. Το δεύτερο στάδιο της αξιολόγησης διεξήχθη μετά την περάτωση του συνόλου του τεχνολογικού πλαισίου, ούτως ώστε να αξιολογηθεί η χρησιμότητά του σε κάθε ομάδα χρήσης. Σε αυτό το στάδιο, η βάση χρηστών της αξιολόγησης διευρύνθηκε, περιλαμβάνοντας παιδιά τριών έως έξι ετών, τους γονείς τους καθώς και ειδικούς στην ανάπτυξη των μικρών παιδιών.

Abstract

Ambient Intelligence (AmI) is a vision of the future information society stemming from the convergence of ubiquitous computing, communication and intelligent user-friendly interfaces. Many AmI applications aim to improve and enhance everyday living activities for a variety of target user groups, including non-traditional users of interactive technologies. However, the potential benefits and impact of AmI technologies for children is underinvestigated. This work aims to build an Ambient Intelligence (AmI) environment which is capable of supporting the development of young children through playing. In order to achieve this goal, this thesis proposes an AmI technological framework to support the design, development and deployment of innovative games. These games are capable of monitoring and evaluating children's skills and abilities, while, on the other hand, enhancing children playing experience as they adapt to meet the continuously changing playing maturity. At the same time, children's needs for activity, exercise and pleasure are optimally covered while also providing opportunities for creativity.

The proposed technological framework facilitates the creation of smart games that share unique features such as: a) being adaptive to children's skills and abilities, b) being able to establish the appropriate communication channels with children (verbal or non-verbal) c) allowing children to have the control so that with practice they are able to play unsupported and thus develop their self-esteem and independence, d) improving motivation, and f) reducing fear of failure.

For producing the proposed technological framework, the stakeholders (i.e. young children, their parents as well as early intervention professionals) were initially identified and their requirements were collected and analysed. This resulted into the elaboration of functional and non-functional requirements allowing the definition of interaction scenarios within the context of AmI environments. Regarding the framework implementation, a service-oriented architecture was conceived, along with the elaboration of various subsystems. The various components were designed and built based on a generic architecture, so as to support knowledge management annotation and extraction, alternative ambient interaction techniques, personalization and adaptation. Furthermore, the proposed framework facilitates the automated extraction of knowledge regarding children's skills, abilities and overall development based on interaction monitoring, so as to offer indications regarding the children developmental state, maturity level and skills. As a result, the provided technological infrastructure allows the detection of potential developmental issues to be further investigated and diagnosed if necessary. To this end, the parents are provided with general information (in a pleasant and practical way) about their child's physical and mental development progress, as well as indications of a possible skill immaturity. Finally, early intervention professionals are provided with extensive data in addition to the full interaction history for reasoning about whether the child is meeting all the necessary developmental milestones. At the same time, this technological infrastructure is employed with the help

of occupational therapists to create innovative ways to use new technology in their practice.

In the context of an AmI environment, the practical usage of the framework has been validated through the design and implementation of a number of augmented artifacts to act as physical or digital means facilitating interaction between children and the system. These artifacts include an augmented interactive children's table, an augmented chair, a digital pen, digital dice, etc. Furthermore, several physical artifacts such as puzzle pieces, identity cards, toys, etc., that typically contain no technology, were augmented using the framework. For the provision of augmented interaction channels several applications were created including a three-dimensional virtual playfellow, digital reproductions of famous games such as puzzles, card games, etc. These applications act as the required software layer to host the novel interaction channels proposed by this research work. For both parents and early intervention professionals to administer these applications and the system, several content editing and interaction monitoring tools were implemented to support each stakeholder group's needs based on the extracted requirements. Finally, the framework itself is a ready to use tool for developers who wish to develop novel applications to support playful interaction within AmI environments. In this respect, the framework provides a number of tools for the creation, interaction scripting, monitoring and integrating of smart augmented artifacts, so as to radically decrease the required development effort.

The aforementioned framework was deployed within a simulation space in the FORTH's AmI Facility and a two-phase evaluation with children, parents and early intervention professionals was conducted in order to identify potential usability barriers prior to the practical exploitation of the concepts. The first phase of the evaluation was conducted early on and after implementing the basic technological infrastructure, so as to primarily measure user acceptance as well as the ergonomic design of the settled artifacts. Participants of this evaluation were children aged from three to six years old. The second phase of the evaluation was conducted after the complete deployment of the framework, in order to assess its value for each targeted stakeholder. Therefore, the evaluation user base was expanded, including young children in the age range of three to seven years old, their parents and early intervention professionals.

Table of Contents

Table of Figures

Table of CodeBlocks

List of Tables

“This Page Intentionally Left Blank”

1 Introduction

1.1 ICT for young children

According to [40], children are currently a substantial segment of the market for Information and Communication Technology (ICT) products and services in Europe. It is estimated that this market will be worth over €30B in a few years. Informal evidence suggests that the most popular Christmas present this year for preschool children was a mobile phone. In schools throughout the EU, children under 12 years of age use the Web every day.

As mentioned in [155], ICT has the potential to provide novel opportunities for children to develop and be creative, to hone generic learning skills and aptitudes and to practice their social skills. Some practitioners, academics, policy-makers and commentators have embraced the use of ICT for children [5], [74], [207]. According to these individuals and organizations, ICT is categorized as a new tool that could and should be incorporated into existing early-years practice in developmentally appropriate ways, supplementing, not replacing, other important first-hand experiences and interactions and accompanied by quality adult input to help children learn about and through the technology [192], [226]. In addition, by introducing state of the art ICT technologies, play can be expanded and enhanced. This is achieved, for example, while children use innovative equipment such as floor robots, walkie-talkies or computers. ICT therefore, is seen as offering a range of potentially valuable pedagogic tools when properly utilized [43], [193].

1.2 ICT for early intervention professionals

Nowadays, the use of ICT is gaining increasing importance in occupational therapy. According to [33], [158], e-health has the potential to change the practice of occupational therapy and will alter the nature and practice of therapeutic relationships, presenting new challenges for the profession. To prepare for these changes, intervention professionals need to be able to utilize existing and emerging ICTs.

Intervention services help children develop different skills including physical, communication, and cognitive skills, and support self-help, as well as social, emotional development and behavioral development. They help children achieve better outcomes, support families, and save precious resources for the community.

Occupational therapists provide support to a wide range of people with physical, psychological or developmental injuries or disabilities. They work with clients across the lifespan, from infancy to old age, with the aim to promote, develop, restore and maintain the abilities needed to cope with daily activities to prevent dysfunction and advance health [242]. Occupational therapists need to be aware of, and involved in, current developments and initiatives in e-health and understand the impact and potential of new technologies on the practice of occupational therapy and what this means for their clients as well as their profession.

According to [35], early intervention occupational therapy practitioners use tele-health technologies to provide services for children. Benefits include increased access to occupational therapy services for children who live in remote areas, the prevention of unnecessary delays in receiving services, and coordinated care among team members in different locations.

A distinction is made between the information and communications technologies that are used by occupational therapists to assist their work (for example, office word processing and databases, email, etc.), and the use of ICT with clients for educational and therapeutic purposes. Technological advancements continually influence practitioner's current practice and occasionally they create new tools for intervention. Virtual rehabilitation is one of these areas of advancements, where changes have driven new and unique treatment methods. According to [196], some practitioners are already using gaming devices, such as the Nintendo Wii, to provide rehabilitative activities, and there are initial indications that technology can assist and enhance occupational therapy practices.

1.3 Ambient gaming and playful interaction

In the last decades, it has been observed that, through a continuously increasing technological progress, game developers have focused on creating more natural and realistic gameplay [200]. A well designed game offers to players an enhanced playing experience. An example of technological progress in ambient gaming is the historic game "PacMan"[2] in comparison with the modern "Call of Duty"[3], both based on the same concept. Despite changes in visuals, gameplay, level design, etc., the interaction (input and output) is still delivered through a physical-controller and a video screen. "Call of Duty" was initially delivered on the PC and later on was expanded to other consoles. These consoles support controllers such as gamepads, joysticks, steering wheels, trackballs, motion sensing and so on, in order to allow a more natural game interaction.

As technological progress expands, significant technological achievements appear in ambient gaming as well. Interactive play environments have been created based not only on the projections of video games and interactive sounds, but also on physiological sensors that measure physiological behavior and other characteristics of the human body. Such characteristics include brain activity, heart rate (i.e. ECG, EEG, EMG, HEG, etc.), respiration (GSR), temperature, iris movements and glucose blood levels. Wearable sensors used in order to extract a variety of measurements include inertia sensors, such as accelerometers, gyroscopes and magnetometers, location sensors, such as GPS and proximity sensors, mini-cameras and muscle tension detectors [200].

[2] http://en.wikipedia.org/wiki/Pac-Man
[3] http://en.wikipedia.org/wiki/Call_of-Duty

The combination of the technological progress both in terms of devices and novel forms of interaction has resulted in the formation of a novel category of games, namely “pervasive”/”ambient” games [60]. These games allow players to act freely in the environment. In addition and according to the properties of Ambient Intelligence [2], qualities for an enriched game experience can include:

- **context-awareness**: (game) devices can recognize players and their situational context
- **personalization**: the provided functionality is tailored to meet players needs and preferences (short timescale, e.g., installing personal settings)
- **adaptation**: the system can change/adapt in response to the people actively engaged in the environment but also the environment itself (adjustments resulting from longer monitoring).

Pervasive and location-based games are another example of games that use Ambient Intelligence technology. These games blend the virtual and real world and support interaction through multiple ubiquitous devices. A location-based game (or location-enabled game) is one in which the gameplay evolves and progresses through a player’s location. Thus, location-based games almost always support some kind of localization technology, for example by using satellite positioning (GPS). Current research trends also use other embedded mobile protocols like Near Field Communication and Ultra Wide Band Wireless (UWB).

As a fundamental human activity, play is engaging regardless of age. At the same time, playful interactions and serious games incorporate elements of play (through their carefully engineered interfaces) to keep interactions sustained and useful for prolonged time periods [190]. The aforementioned technologies can fuel fun and entertainment in AmI applications and create smart engaging playful interactions with end users, and especially young children.

1.4 Young children in AmI environments

The range of ICT available today and foreseen for the near future provide the means to achieve a radical transformation of teaching and learning than desktop computers alone would provide. According to [190], a large number of products are available to young children that incorporate some aspect of ICT. These include activity centers, musical keyboards, tape recorders, programmable and radio-controlled toys as well as everyday items such as remote controls, telephones, fax machines, televisions and computers. This range of toys and devices is part of the move towards pervasive or ubiquitous computing in which technology blends into the environment and is not necessarily visible.

In computing, ambient intelligence (AmI) refers to electronic environments that are sensitive and responsive to the presence of people. According to [1], the ambient intelligence paradigm builds upon pervasive computing, ubiquitous computing, profiling, context awareness, and human-centric computer interaction design. In this context, AmI environments offer opportunities for supporting the learning needs of children and integrate ICT into teaching situations at home and at schools in a variety of ways.

Having in mind that play is the dominant medium for learning in pre-school education and as mentioned earlier, games can be effective tools for pedagogy, the following step is the creation of appropriate innovative games to support children's learning and development.

Designing and creating games under the perspective of Ambient Intelligence has the potential to provide enhanced playing experience to all, and in particular to children. Such games are facilitated by systems and technologies that are:

- **embedded**: games are integrated into the environment
- **context aware**: they can recognize children and their situational context
- **personalized**: they can be tailored to their needs
- **adaptive**: they can change in response to young children
- **anticipatory**: they can anticipate children's desires without conscious mediation.

1.5 Contribution

Play helps children to enhance and develop fundamental skills, such as physical, social, emotional, intellectual, creative skills, etc. Taking into consideration the fact that play has a vital role in the everyday life of a child, the work presented in this thesis aims to build an Ambient Intelligence environment for supporting the development of young children through playing. In detail, the aim of this research work is to provide an AmI technological framework for supporting the design, development and deployment of smart child development games. The aim of such games is twofold. Firstly, they are capable of monitoring and evaluating children's skills and abilities, while, on the other hand, they enhance the child's playing experience as they adapt to meet the continuously changing child's playing maturity. Smart child development games cover part of the child's needs for activity, exercise and pleasure, as well as provide opportunities for creativity encouragement.

To this end, the contribution of this work is the following:

- An adaptation infrastructure mechanism for the automated extraction of child development knowledge, based on interaction monitoring, targeted to model relevant aspects of children developmental state, maturity level and skills.
- A new scripting language, called ACTA, for facilitating the activity analysis process during smart game design by early intervention professionals. Furthermore, developers can use ACTA not only for developing event-driven sequential logic games, but also for applications of behavior composed of a finite number of states, transitions between those states, and actions as well as for application based on rules driven workflows.
- A child monitoring framework based on occupational therapy's expertise aiming at early detection of children's potential delays to be further investigated and diagnosed if necessary.

- A pool of novel interaction metaphors and techniques and the appropriate administration and training facilities. To achieve this objective, ICT technology was used so as to capture and intelligently transform user interaction data. The result was the production of an AmI interaction framework that offers among other gesture/posture recognition based on the dynamic time warping algorithm (DTW).
- A collection of augmented artifacts, including an augmented children's table, an augmented chair, a digital pen, digital dice, etc. The implementation of these artifacts was conducted taking into account the basic principles of AmI and ubiquitous computing, thus making devices unobtrusive and technology hidden or embedded in traditional everyday objects and furniture. General health and safety requirements for children were also taken into account.
- Transformation of common physical artifacts such as puzzle pieces, identity cards, toys, etc., using computer vision technics.
- A novel, remotely controlled three-dimensional full body avatar framework, for the purposes of edutainment and instructor-student interaction. The main innovation focuses on multi-presence gamified educational scenarios in multiple desktop computers and mobile devices.
- Content editing and interaction monitoring tools supporting each stakeholder group based on the extracted requirements.
  - Parents are provided with general information (in a pleasant and aesthetic way) about their child's physical and mental development progress, as well as indications of a potential skill immaturity.
  - Early intervention professionals are provided with extensive data in addition to the full interaction history for reasoning about whether the child is meeting all the necessary developmental milestones.
- A collection of digital reproductions of famous games such as puzzles, card games, labyrinths, the tower game, the farm game, the mimesis game, etc. These games were designed with the assistance of HCI experts and occupational therapists.

2 Background and Related work

This work aims to build an Ambient Intelligence environment for supporting the development of young children through playing. In detail, the aim of this research work is to provide an AmI technological framework for supporting the design, development and deployment of smart child development games. This section establishes the foundations of this research work by identifying the state of the art in the targeted application domains.

2.1 Ambient Intelligence

The term Ambient Intelligence (AmI) refers to electronic environments that are sensitive and responsive to the presence of people [164]. In 2001, the Information Society Technologies Advisory Group (ISTAG) proposed a longer-term perspective in preparation of the next European Community Framework Program for Research and Technological Development [53]. With the help of experts from across Europe, a vision was elaborated describing what living with 'Ambient Intelligence' might be like for ordinary people in 2010 [101].

According to ISTAG [102], the concept of AmI provides a vision of the Information Society where the emphasis is on greater user-friendliness, more efficient services support, user-empowerment, and support for human interactions. People are surrounded by intelligent intuitive interfaces that are embedded in all kinds of objects and an environment that is capable of recognizing and responding to the presence of different individuals in a seamless, unobtrusive and often invisible way.

The vision of AmI assumes a shift in computing from desktop computers to a multiplicity of computing devices in our everyday lives, whereby computing moves to the background and intelligent, ambient interfaces to the foreground (see Figure 1).

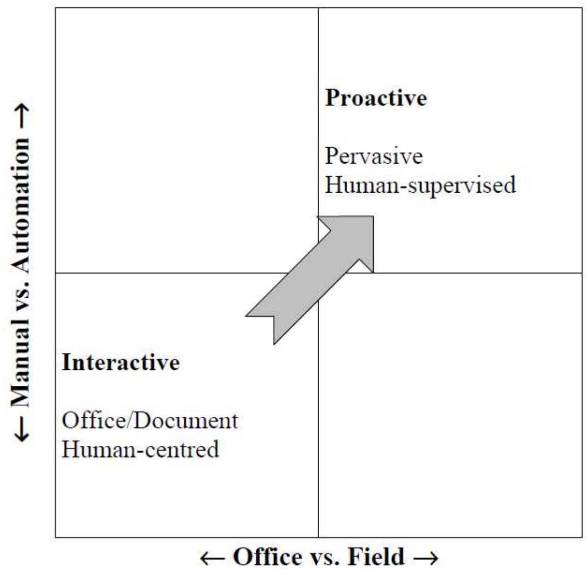


Figure 1: The shift from interactive to proactive computing

In the context of Ambient Intelligence (AmI), the elaboration of new interaction techniques is becoming the most prominent key to a more natural and intuitive interaction with everyday things [1]. Natural interaction between people and technology can be defined in terms of experience: people naturally communicate through gestures, expressions, movements. To this end, people should be able to interact with technology as they are used to interact with the real world in everyday life [228]. Additionally, AmI systems must be sensitive, responsive, and adaptive to the presence of people [169].

More specifically, Ambient Intelligence (AmI) presents a vision of a technological environment capable of reacting in an attentive, adaptive and active (sometimes proactive) way to the presence and activities of humans and objects in order to provide appropriate services to its inhabitants [213]. According to the Institute for the Future [26], emerging technologies are transforming everything that constitutes our notion of “reality” - our ability to sense our surroundings, our capacity to reason, and our perception of the world. In the context of Ambient Intelligence, several challenges emerge in the contributing domains of ubiquitous computing, mixed reality and Human Computer Interaction (HCI).

The term “Ambient Intelligence” was coined from Philips Research’ vision of “people living easily in digital environments in which the electronics are sensitive to people's needs, personalized to their requirements, anticipatory of their behavior and responsive to their presence” [167]. This concept was adopted by the Information Society Technologies Advisory Group (ISTAG) as one of their research focus. In their report [102], ISTAG shows the concrete vision that humans will, in an Ambient Intelligent Environment, be surrounded by intelligent interfaces embedded in everyday objects (as furniture, clothes and the environment). According to [2] and [3], the five key technology features that portray an AmI system are:

- **Embedded**: many network devices integrated into the environment. The environment system can judge the situation from the device input, and backend devices share information with the environment system to support users in physical space.
- **Context aware**: the environment can determine the context in which certain activities take place, where context relates meaningful information about persons and the environment, such as positioning and identification.
- **Personalized**: the environment can be tailored to the individual needs of users. It can recognize users and adjust its appearance to maximally support them. Automatic user profiling can capture individual user profiles through which personalized settings and information filtering can be accommodated.
- **Adaptive**: the environment can change in response to the users’ needs. It can learn from recurring situations and changing needs, and adjust accordingly.
- **Anticipatory**: the environment can act upon the user’s behalf without conscious mediation. It can extrapolate behavioral characteristics and generate pro-active responses.

These technology features primarily facilitate intelligent communication while interacting with AmI environments. Moreover, gestures are also employed in the context of AmI for providing alternative ways of user input in a human like fashion. Research conducted in this field involves the usage of gestures for providing input to augmented desk interface systems using multiple fingertips recognition (identify fingertips and their trajectories and infer gestures based on these trajectories) [157] [150]. Computer vision is used for identifying hand gestures, facial expressions and body postures [156]. Furthermore, the usage of thimble-shaped fingertip markers made of white printing paper with a 'black light' source, has been proposed for providing gesture recognition in the context of back projection walls [114]. Although this form of interaction is not generically applicable, it can be used in conjunction with face tracking and head position estimation to support various forms of natural implicit interaction with the environment.

In addition, research has been conducted targeted to using speech as an input channel in ICT for years, and today thousands of commercial and research product are available.

AmI is comprised of three main components: ubiquitous computing, ubiquitous communication, and user adaptive interfaces, also referred to as "Intelligent social user interfaces" (ISUIs) [125]. These interfaces go beyond the traditional keyboard and mouse to improve human interaction with technology by making it more intuitive, efficient, and secure. They allow the computer to know and sense far more about a person, the situation the person is in, the environment, and related objects than traditional interfaces can.

Intelligent social user interfaces (ISUIs) encompass interfaces that create a perceptive computer environment rather than one that relies solely on active and comprehensive user input. ISUIs can be grouped into the following categories:

- Visual recognition (e.g. face, 3D gesture, and location) and output
- Sound recognition (e.g. speech, melody) and output
- Scent recognition and output
- Tactile recognition and output
- Other sensor technologies.

Traditional user interfaces like PC-controlled touch screens and user interfaces in portable units such as PDAs or cellular phones can also become ISUIs. The key to an ISUI is the ease of use, in this case the ability to personalize and adapt automatically to particular user behavior patterns (profiling) and different situations (context awareness) by means of intelligent algorithms. In many cases, different ISUIs, such as voice recognition and touch screen, are combined to form multi-modal interfaces [7].

2.1.1 Driving forces

Intelligent User Interfaces are becoming increasingly important as users face increasing system complexity and information overload [136]. These interfaces are adaptive to heterogeneous user populations, employing the full power provided through modern user and context modelling facilities. At the same time, the environment itself is changing; microprocessors are integrated into everyday objects like furniture, clothing, white goods and toys. This section describes how Ubiquitous Communication enables these objects to communicate with each other and the user by means of ad-hoc and wireless networking [164]. Furthermore, Mixed reality (MR) (encompassing both augmented reality and augmented virtuality) refers to the merging of real and virtual worlds to produce new environments and visualizations, where physical and digital objects co-exist and interact in real time. Last but not least, Augmented Reality (AR) allows virtual imagery to augment and enhance physical objects in real time. In this new environment machines are more than ever becoming anticipatory and responsive to human needs.

2.1.1.1 Ubiquitous Computing

Ubiquitous Computing has as its goal to enhance computer use not only by making many computers available throughout the physical environment, but also by making them effectively invisible to the user [233]. The idea of ubiquitous computing was first thought by Mark Weiser in 1998 at the Computer Science Lab at Xerox PARC[4]. He envisioned computers embedded in walls, tabletops, and everyday objects. A person might interact with hundreds of computers at a time, each invisibly embedded in the environment and wirelessly communicating with each other [234]. A number of researchers around the world are now working in the ubiquitous computing framework. Their work impacts all areas of computer science, including hardware components (e.g., chips), network protocols, interaction substrates (e.g., software for screens and pens), applications, privacy, and computational methods. Some researchers say that ubiquitous computing is the Third Wave of Computing [183]. The First Wave was "many people, one computer", and the Second Wave, the PC, is the era of "one person, one computer". The Third Wave will be the era of many computers per person.

According to [58], ubiquitous or pervasive computing assumes a large number of 'invisible' small computers embedded into the environment and interacting with mobile users. Users will experience the world through devices to wear (e.g., medical monitoring systems), to carry (e.g., personal communicators that integrate mobile phones and PDAs), devices that are implanted in the vehicles or the public spaces (e.g., car and public space information systems), and devices integrated in the architectural environment (e.g., interactive walls and furniture). This heterogeneous collection of devices will interact with intelligent sensors and embedded actuators in homes, offices, public spaces and transportation systems, in order to form a mobile ubiquitous computing environment, which aids normal activities related to work, education, entertainment and healthcare. The environment will also provide access to wired backbone computing resources, connected to the Internet.

The **Disappearing Computer initiative** had as an overall goal to explore how everyday life can be supported and enhanced through the use of collections of interacting artifacts. Together, these artifacts form new people-friendly environments in which the "computer-as-we-know-it" has no role [214]. The aim was to arrive at new concepts and

[4] http://www.parc.xerox.com

techniques out of which innovative applications can be developed. In order to achieve this goal, three specific interlinked objectives were established:

- Developing new tools and methods for the embedding of computation in everyday objects so as to create artifacts
- Researching how new functionality and new uses can emerge from collections of interacting artifacts, and
- Ensuring that people's experience of these environments is both coherent and engaging in space and time.

In the context of Ubiquitous Computing two types of "disappearance" have been defined, namely physical and mental disappearance [59]:

- **Physical disappearance** is achieved by the miniaturization of computer parts that allows convenient and easy integration into other artifacts, mostly into "close-to-the-body" objects so that the result can fit in your hand. These could be integrated in clothing or even implanted in the body. Features, usually associated with a computer, are not visible anymore and the interaction happens via the compound artifact in which the computer parts disappear.
- **Mental disappearance** of computers is achieved by becoming "invisible" to the "mental" eye of the users. This can happen by embedding computers or parts of computers in the architectural environment (walls, doors, etc.) or in furniture (tables, chairs, etc.). Traditional everyday objects can yet interact, communicate, and cooperate with human due to the multifunctional character they get. An interactive table, for instance, has still the standard table form, though it offers additional functions to the users, because of an embedded computer.

Nowadays, computers are commonly used by everyone in everyday life, but they demand high attention from the users, while on the other hand, there are several technical features that a ubiquitous computing system must have [183]:

- **Terminal and user interface issues**: the devices that are used should have a good display quality and responsiveness to user input. Even with limited display sizes, the use should be as intuitive and clear as possible, while different means of data input such as pens, handwriting recognition or speech recognition should be supported.
- **Low cost devices**: although general purpose computers are more expensive, the kind of computers that will be used for ubiquitous computing does not have to be that expensive, due to the fact that specific computers for concrete applications, for example, do not have the processor and hard disk requirements that general purpose computers have.
- **High bandwidth**: enough network bandwidth is important to allow the communication between different devices. Besides, it is not only required to build a high-bandwidth network that is capable of communicating with the different terminals, but there are also many issues related to the current state of the system like establishing the locations of mobile terminals (if they are used), making the best

use of the available frequencies, maintaining the quality of services, encrypting data, eliminating network latency, etc.

- **'Invisible' file systems**: one of the requirements of ubiquitous computing is that computers should become invisible, and they should be able to understand the user while, for example, voice recognition or other intuitive interfaces are applied.
- **Automatic installation**: ubiquitous computing should eliminate the need of program installation. The concept of installation has no sense in ubiquitous computing, though programs must be able to move from a computer to another without requiring fundamental changes to each computer's configuration. Some technical alternatives are the use of programming languages, such as Java, that are platform-independent and are moved easily from one computer to another.
- **Personalize information**: in order a ubiquitous computing system to personalize the information that presents depending on the user, the profile needs to be added to every device every time a new person joins a community.
- **Privacy issues**: one of the most important problems of ubiquitous computing is that it can generate serious privacy risks. The system can record the actions of the users, their preferences or even their locations, though some other people may access to their personal data. For that purpose, new network technologies such as infrared or wireless radio communication use encryption to ensure security.

2.1.1.2 Augmented and Mixed reality

Augmented Reality (AR) allows virtual imagery to augment and enhance physical objects in real time. Users may interact with the virtual images using real objects in a seamless way [249]. Progress in computer vision largely contributes to innovative interaction in Ambient Intelligence environments through techniques such as image acquisition, image processing, object recognition (2D and 3D), scene analysis, and image flow analysis, which can be exploited for humans' and objects' recognition and tracking [177]. At the same time, ICT components are embedded into everyday objects like furniture, clothing, white goods, toys, etc., [8]. Augmented objects can be used for providing implicit or explicit input to systems while their physical and mental existence as computational devices disappears [215]. Ambient interaction merges real and virtual worlds to produce new environments and visualizations where physical and digital objects co-exist and interact in real time. Additionally, in Ambient Intelligence environments interaction is monitored and implicit input is also extended to include empathy to understand human's feeling or states.

Mixed reality (MR), encompassing both augmented reality and augmented virtuality, refers to the merging of real and virtual world in order to produce a new environment and visualization, where physical and digital objects co-exist and interact in real time. In other words, mixed reality is a mix of reality, augmented reality, augmented virtuality and virtual reality [143].

Mixed reality (MR) refers to the incorporation of virtual computer graphics objects into a real three dimensional scene, or alternatively the inclusion of real world elements into a virtual environment. The former case is generally referred to as augmented reality, and the latter as augmented virtuality. Three characteristics that are integral to an augmented reality interface have been defined [14]. Firstly, it combines the real and the virtual. Secondly, it is interactive in real time and thirdly, it is registered in three dimensions.

Single user Mixed Reality interfaces have been developed for computer aided instruction [62], manufacturing [45] and medical visualization [15]. These applications have shown that Mixed Reality interfaces can enable a person to interact with the real world in a way that was possible never before. As an instance, relative work overlays virtual ultrasound images onto a patient's body, allowing doctors to have "X-Ray" vision while performing a needle biopsy task [15].

Moreover, the design of mixed reality systems gives rise to new challenges due to the new roles that physical objects can play in an interactive system. In addition to the design of mixed objects, interacting within such mixed environments composed of physical, mixed and digital objects, involves novel interaction modalities and forms of modalities that require new interaction models.

In particular, Mixed Reality can be implemented in various sectors, such as computer-aided design [54], arts, games, learning and entertainment [31] [153]. With respect to learning, Mixed Reality has the potential to:

- **Enable constructivism and experiential learning**: virtual environments are unique in their usefulness to education due to their characteristics of autonomy, presence and interaction [239]. These are in line with important paradigms in learning, such as experiential education and constructivism [29].
- **Enables social or collaborative learning**: social learning is one of the key learning theories mentioned in [29]. Multi-user virtual environments and augmented reality could help meet the need for social learning [47]. Collaboration is considered useful to the learning process because students have to articulate and debate their positions, thereby leading to reflection and co-constructed knowledge [104].
- **Presence**: allowing students to feel a sense of presence in the virtual world is one aspect that enables VR learning systems to increase motivation [144]. Research in the domain of educational virtual environments [142] has provided feedback that factors such as environmental richness and a high level of interactivity, contributed to a high degree of presence, which was in turn associated with knowledge construction.
- **Interactivity**: interactivity is generally seen as an intrinsic feature of educational practice in the sense of social communication, but also as an inherent property of any interactive multimedia or virtual reality environment that promises physical and sensor, in addition to mental, activity and response [187]. According to [216], interactivity is examined in two ways - the first is a generic view relating to the frequency of interaction, the range of choices when interacting with the system and the significance of those choices; the second, specific to virtual environments, is defined as the tight coupling of head-tracked (and other) user input with multisensory display, so users perceive themselves as present in the environment.

> As to the importance of interactivity, it is argued [187] that there was a strong relationship between interactivity, engagement and learning.

It is also important to mention that Mixed Reality has been employed for education in a number of ways. The Web3D [124] educational tool, for instance, allows students to learn engineering related knowledge through a website in AR mode. Several educational exhibits for science centers, museums and education centers such as BlackMagic Kiosk, Solar System and Story Book have been also set up [241]. Additionally, a tangible interface for storytelling using gesture recognition and collaborative navigation technologies has been designed, as a result of the collaboration between researchers, children and teachers [211]. Using the latter interface, children can create their own stories. Moreover, Construct3D [110] is a system used for mathematics and geometry education, while researchers from MIT proposed a new concept, called Games to Teach[5] and they have developed three prototypes for electromagnetic, environment and history educations.

Figure 2: Augmented Reality in Entertainment

As augmented reality methods are increasingly developed in a variety of fields, nowadays many applications aim to entertain, as well. Users can navigate, in real time and in 3D, when having a smartphone or a tablet as in the meantime they follow instructions in order to find the right path, while collecting bonus gifts or treasure. Users are able to see these collectables in the mobile screen through the phone/tablet camera even though the object doesn't exist on real world. Users can look around for the collectables and grab them to redeem offers and prizes, as depicted in Figure 2.

[5] http://icampus.mit.edu/projects/GamesToTeach.shtml

2.1.1.2.1 Annotation and visualization

AR could be used to annotate objects and environments with public or private information. Applications using public information assume the availability of public databases to draw upon. For example, a hand-held display could provide information about the contents of library shelves as the user walks around the library [66] [179] [178]. Google has, furthermore, recently released a new invention based on Ubiquitous Computing, known as "Google Glasses" (see Figure 3Figure 3). It is about a multifunctional platform which aims to enhance the environment and objects that surround the user giving him information he needs like temperature, descriptions of a specific object etc., [78].

Figure 3: Augmented Reality through Google glasses

2.1.2 Intelligent User Interfaces

A user interface has to make several communication decisions while interacting with the user. These decisions may concern several aspects of the interaction, such as ***what*** to communicate, ***when*** to communicate, and ***how*** to communicate. According to [221], a user interface is intelligent to the extent that it adapts itself, and makes these communication decisions dynamically, at run-time, based on the requirements of the interaction. The area of intelligent interfaces covers a variety of topics concerned with the application of Artificial Intelligence and knowledge-based techniques to issues of human-computer interaction.

Computer applications grow increasingly complex and through the use of artificial intelligence technology software systems begin to achieve the ability to reason and make decisions on their own. Consequently, the design of effective and efficient human-computer interfaces becomes ever more critical to overall system performance. Advanced applications are characterized by large amounts of information to be conveyed and understood, complex task structures, real-time performance characteristics, and incorporation of autonomous or semiautonomous agents. The requirements imposed by human-computer interaction with such systems exceed the capabilities of conventional interfaces which often fail to reflect the semantics of its users' tasks and problem domain properly. Intelligent User Interfaces (IUIs) aim to cope with these serious semantic problems and help users to access information or solve complex tasks by being sensitive to a user's knowledge, misconceptions, goals, and plans [32], [39], [135], [219].

The main issues addressed by intelligent user interface research are the following:

- How can interaction be made clearer and more efficient?
- How can interfaces offer better support for their users' tasks, plans, and goals?
- How can information be presented more effectively?
- How can the design and implementation of good interfaces be made easier?

The most important property of intelligent user interfaces is that they are designed to improve communication between the user and machine. It is not important what kind of technique is being used to achieve this improvement, as long as it can be regarded as “intelligent”. According to [165], the types of techniques that are applied in intelligent user interfaces are:

- **Intelligent input technology:** uses innovative techniques to get input from a user. These techniques include natural language, gesture tracking and recognition, facial expression recognition, and gaze tracking among others
- **User modeling:** covers techniques that allow a system to maintain or infer knowledge about a user based on the received input
- **User adaptivity:** includes all techniques that allow the communication between human and machine to be adapted to different users and different situations for example, machine learning or context awareness
- **Explanation generation**: covers all techniques that allow a system to explain its results to a user for example, speech output, intelligent interface agents and tactile feedback in a virtual reality environment.

Besides improved communication, other important properties of IUIs are personalization and flexibility of use. To achieve personalization, IUIs often include a representation of the user. These user models log data about the user’s behavior, knowledge, and abilities. New knowledge about the user can be inferred based on the input and interaction history of the user with the system.

In order to be flexible, many IUIs employ adaptation or learning. Adaptation can occur based on the stored knowledge in a user model or by making new inferences using current input. Learning occurs when stored knowledge is changed to reflect new data. Because of the difficulties involved in creating IUIs and the amount of knowledge engineering that is needed, most IUIs focus on a specific method of interaction (e.g. speech) or on a particular well-defined application domain.

2.1.2.1 User and context profiling

The scope of user profiling is to provide information regarding the user currently accessing an interactive application. In this context, a user profile initially contains attributes specified by the user prior to the initiation of interaction or during interaction (based on interaction monitoring). On the other hand, context-profiling aims at collecting context attribute values related to [140]:

- The environment, i.e., the physical space where the user is located, relationships to other people (people in the same neighborhood, people connected to by the communication services), the position of the user, the direction and speed etc.
- The situation the user is in (emergency situation, someone's birthday, a car accident, etc.)
- The activity of the user
- The role of the user
- The service the user has access to and/or is using.

Required characteristics in creating effective user modeling systems have been documented in [73] and [117]:

- **Generality, including domain independence:** user modeling systems should be usable in as many domains as possible, and within these domains for as many user modeling tasks as possible.
- **Expressiveness and strong inferential capabilities:** expressiveness is a key factor in user modeling systems; they are expected to express many different types of assumptions about the users and their context. Such systems are also expected to perform all sorts of reasoning, and to perform conflict resolution when contradictory assumptions are detected.
- **Support for quick adaptation:** time is always an important issue when it comes to users. User modeling systems are required to be adaptable to the users' needs. Hence, they need to be capable of adjusting to changes quickly.
- **Precision of the user profile:** the effectiveness of a user profile depends on the information the system delivers to the user. If a large proportion of information is irrelevant, then the system becomes more of an annoyance than a help. This problem can be seen from another point of view: if the system requires a large degree of customization, then the user will not be willing to use it anymore.
- **Extensibility:** a user modeling system's success relies on the extensibility it offers. Companies may want to integrate their own applications (or API) into the available user models.
- **Scalability:** user modeling systems are expected to support many users at the same time.
- **Import of external user-related information:** user models should support a uniform way of describing users' dimensions in order to support integration of already existing data models.
- **Management of distributed information:** the ability of a generic user modelling system to manage distributed user models is becoming more and more important.

Distributed information facilitates the interoperability and integration of such systems with other user models.

- **Support for open standards:** adherence to open standards in the design of generic user modelling systems is decisive since it fosters their interoperability.
- **Load balancing**: user modelling servers should be able to react to load increases through load distribution and possibly by resorting to less thorough (and thereby less time-consuming) user model analyses.
- **Failover strategies:** centralized architectures need to provide fallback mechanisms in case of a breakdown or unexpected situation.
- **Fault tolerance:** in case a user inserts wrong data in his/her profile by mistake (i.e. a user denotes an opposite gender), the system must prompt the user to adjust the corresponding parameters, rather than reset his/her profile.
- **Transactional Consistency:** parallel read/write procedures on the user model should lead to the deployment of sufficient mechanisms that preserve and restore possible inconsistencies.
- **Privacy support:** another requirement that user modelling systems must meet in order to become acceptable by the general public is to respect and retain the user's privacy. In order to meet these requirements, such systems must provide a way for the users to express their privacy preferences, as well as the security mechanisms to enforce them.

On the other hand, effective context modelling systems must be able to [55]:

- **Gather context information of different types**: the technique should be flexible enough to handle different kinds of context information and should not be limited to any specific type of context information, for e.g. information about a user's location from various location sensors.
- **Gather context information without end user intervention:** the technique should be able to be used in self-supporting or self-learning environments and should not require explicit end user input, for e.g. downloading the users' interests from their personal web pages instead of inserting from scratch all their profiles' parameters.
- **Aggregate context information from potentially multiple sources**: the technique should enable the aggregation of context information from potentially multiple sources, for e.g. accessing various URIs to download and aggregate all users' profile parameters.
- **Predict context behavior:** the technique should be able to derive additional information from the information in the context information representation, such as prediction of context behavior. For e.g., if a user deviates too often then the system has to adjust her routing preference accordingly, in order to be conformed to her profile.
- **Adjusting behavior in a certain (pre-defined) situation:** the technique should be able to describe behavior (next to context) and define rules when certain behavior (action) should be executed. For e.g., if a user deviates, then an appropriate rule should be fired in order to force dynamic path-replanning.

- **Handle different representations of context information:** the technique should be able to handle different representations of context information, for e.g. dealing with different calendar formats.

2.1.2.2 Approaches to user and context modeling

2.1.2.2.1 Static profiling

Static profiling entails the complete specification of attributes prior to the implementation of the reasoning engine of an interactive application. Where static profiling is employed, the process of altering the logic used for generating the adaptable behaviors of the system is semi-automatic and cannot occur on the fly. More specifically, it is not feasible, when such an approach is followed, to enrich the decision logic while the system is running to perform meta-adaptation. This can only be achieved in the context of adaptations that occur based on collecting and analyzing usage data.

2.1.2.2.2 Extensible profiling using special purpose languages and design support tools

A potential solution to the issues that arise when using static profiling is to separate the logic under which adaptations occur from the system performing the adaptation. This is achieved through the creation of special purpose languages for the specification of the decision logic. An example of such a language is DMSL [194]. When such languages are used, the design of an application in term of alternative styles and their activation can be carried out using special purpose design support tools, such as MENTOR [11], which can be used to produce the decision logic of an application and therefore orchestrate the user interaction.

2.1.2.2.3 Extensible profiling using semantic data modeling

Recent approaches to information representation have made possible the creation of a profiling scheme using a knowledge base modeled with the help of a web ontology language such as OWL to store the appropriate information in the form of semantic web rules and OWL-DL [95] ontologies. This approach has many advantages, as it offers enough representational capabilities to develop a formal context model that can be shared, reused, extended for the needs of specific domains, but also combined with data originating from other sources, such as the Web or other applications. Moreover, the development of the logic layer of the Semantic Web is resulting in rule languages that enable reasoning about the user's needs and preferences and exploiting the available ontology knowledge [141].

2.1.2.3 User interface adaptation

In computing, the notion and importance of *adaptation*, as the ability to adapt a system to the user's needs, expertise and requirements was only recently recognized. In a natural environment, the survival of all living organisms is often, if not constantly, subject to their ability of constantly changing, adjusting and functioning according to the surrounding environment. In similar terms, the "survival" of users (in other words, system adoption) could also be considered as subject to: (a) their ability to adapt to a given system (environment), on the one hand, and (b) to their freedom to modify the settings of the system in question, on the other hand. Clearly, early forms of UIs were rather based on the first case alone. This was mainly due to the fact that early interfaces had to be taken as de facto, restricting their users to make convenient changes to the style, presentation and behavior of a given UI. A radical change in this pattern has recently emerged with the introduction of adaptation in user interfaces: the computationally empowered environment can adapt itself, at various degrees, to its 'inhabitants', thereby reducing drastically the amount of effort required from the user's part. Taking into account the principles of iHCI and employing the information that Ambient Intelligence can provide regarding the current situation, it becomes possible to build user interfaces that employ adaptation to answer user requirements through implicit or explicit input. At a very general level, the requirements for the User Interface (UI) are dependent on the application, the UI hardware available, the user and the context [198]. The requirements defined by the application may be quality parameters for the visualization of certain content. The UI can be a single device with specific properties or a distributed configurable UI system with various input and output options. In this context, some preliminary adaptation issues include [198].

- **UI adaptation for distributed settings**: in environments where there is a choice of input and output devices, it becomes central to find the right input and output devices for a specific application in a given situation. In an experiment where web content, such as text, images, audio-clips, and videos are distributed in a display rich environment, for example, the context is a key concept for determining the appropriate configuration [55].
- **UI adaptation in a single display**: adapting the details in a single user interface at runtime is a further big challenge. Here in particular adaptation of visual and acoustic properties according to a situation is a central issue. Simple examples that are by now available in different commercial products are the adjustment of the volume according to the environmental sound level and the regulation of backlight depending on the ambient light level. Experiments have been conducted in order to evaluate the concept of having fonts and font sizes in a visual interface that are dependent on the situation. Mainly dependent on the user's activity the size of the font was changed. In a stationary setting the font was small whereas when the user was walking the font was made larger to enhance readability [55]. The orientation aware display described in [199] belongs also in this category.

2.1.2.3.1 User interface adaptation toolkits (adaptable user interfaces)

Data stemming for user and context profiling are used by adaptation toolkits for dynamically generating the interface instance that is more appropriate for the specific user and under the specific context of use. These frameworks in their most advanced implementation consist of a collection of alternative interaction elements mapped to specific user and context parameters. The automatic selection of the appropriate elements is the key for supporting an exponential amount of alternative interface instantiations.

EAGER [51] is a development toolkit that allows Web developers to build adaptive applications using facilities similar to those offered by commonly user frameworks (such as ASP.NET[6] and Java server faces[7]). It is a developer framework; build over ASP.NET providing adaptation enabled ready to use dialogs. By means of EAGER, a developer can produce Web portals that have the ability to adapt to the interaction modalities, techniques and UI elements most appropriate to each individual user, according to profile information containing user and context specific parameters.

Another similar toolkit, reported in [123], aims to facilitate the implementation of adaptive-aware user interfaces for mobile services. UI widgets supported by this framework encapsulate all the necessary information and are responsible for requesting and applying the relative decisions. The toolkit employs DMSL to allow UI developers to turn hard-coded values of lexical attributes to adapted UI parameters specified in an external preference file. As a result, the UI Implementation is entirely relieved from adaptation-related conditionality, as the latter is collected in a separate rule file.

2.2 Technologies for Ambient Intelligence

This section describes which fundamental technologies can be implemented in Ambient Intelligence systems and how smart artifacts enhance AmI environments. Application examples are additionally presented for each case of the above mentioned.

2.2.1 Wireless technologies

At this point, the various technologies that contribute in the completion of an Ambient Intelligence environment are described, while obtaining a flexible communication system.

Different wireless technologies, as presented below, are available on the market or currently under development [6]. The main factors to be considered when selecting the technology are physical size, range of operation, data transfer rate, and price. Physical size may become a critical factor due to space constraints, especially in mobile applications. Following Moore's law, wireless hardware has been decreasing in size considerably and can nowadays be placed on small microchips. The University of California, Berkeley demonstrated this by building a wireless sensor only 2.0 by 2.5 mm in size that can transmit data at a speed of 19,200 Kbit/s over a range of up to 40 feet [248].

- **Wireless LAN (W-LAN):** applications per standard IEEE 802.11b offer high-speed transfer rates of 11 Mbit/s and can be extended over entire office buildings and production areas by using several access points. While W-LAN is considerably cheaper than a traditional stationary LAN, it is often still too costly to be included in small individual devices.
- **Bluetooth technology:** is used in today's handheld applications like cellular phones or personal digital assistants (PDAs) per standard IEEE 802.15 to allow wireless connection within a personal area network (W-PAN). While the cost of Bluetooth equipment is significantly lower than the cost of W-LAN, the transmission range of

[6] http://msdn.microsoft.com/en-us/library/ms644563.aspx, ASP.NET Web Applications
[7] http://java.sun.com/j2ee/1.4/docs/tutorial/doc/JSFIntro.html ,JavaServer Faces Technology

up to 10 meters and the data transfer rate of less than 720 Kbit/s are inferior. New Bluetooth versions are currently under development that attempt to eliminate the latter drawback. V1.2 allows rates of up to 3 Mbit/s, V2.0 of up to 12 Mbit/s.

- **High rate W-PANs:** per standard IEEE 802.15 TG3, launched in 2003, use higher power devices (8 dB) than regular Bluetooth equipment (0 dB) to transmit data at a rate of up to 55 Mbit/s and over a range of up to 55 m. This technology is, therefore, an attractive alternative to W-LAN, especially considering the comparatively lower cost.
- **Low power W-PANs:** per standard IEEE 802.15 TG4 are particularly useful for handheld devices since energy consumption for data transmission purposes, and costs, are extremely low. The range of operation of up to 75 m is higher than current Bluetooth applications, but the data transfer rate of 250 Kbit/s is lower.
- **Wireless body area networks (BANs):** interlink various wearable devices, such as wireless data glasses, earpieces, microphones, and sensors, and can connect them to outside networks. BANs are often used for medical applications but also in work-related fields, for example, to provide production operators with instructions that are adapted to the respective work situation. BANs usually consist of a central network unit, which connects the devices and which can provide an interface to further networks outside the BAN, for example, via Bluetooth. Advantages of BANs versus W-PANs are the short range and the resulting lower risk of tapping and interference, as well as low frequency operation, which leads to lower system complexity. Technologies used for wireless BANs include magnetic, capacitive, low-power far-field and infrared connections, while non-wireless BANs use wires or conductive fabrics.

2.2.2 Sensing technologies

Ambient Intelligence is designed for the real-world, therefore the effective use of sensors in physical environments is vital. Physical components allow an intelligent agent to sense and act upon the environment enhancing the computational power. Ambient Intelligence software algorithms rely on sensory data from the real world in order to perceive the environment, use this information to reason about the environment, and determine the action that can be taken to change the state of the environment. Perception is accomplished using a variety of sensors. The most popular use cases of sensors are listed below:

- Position measurement (position, velocity, direction) [240]
- Detection of chemicals and humidity sensing [48]
- Determine readings for light, radiation, temperature, sound, strain, pressure
- Physiological sensing to support health monitoring [59], [210]
- Motion sensors
- Identification sensors (i.e., RFID).

The next section provides and overview of hardware and software sensors, explaining their characteristics and spectrum of use, in association with the aim of the present research work.

2.2.2.1 Hardware sensors

Hardware sensors have a wide range of different attributes and they are typically quite small so that they can be integrated into almost any AmI application. In detail, hardware sensors can include:

- **Accelerometers**[8]**:** they have been implemented in multiple applications in industry and science, for instance, highly sensitive accelerometers are components of inertial navigation systems for aircraft and missiles. Accelerometers are used to detect and monitor vibration in rotating machinery, to obtain that images on screens in tablet computers and digital cameras are always displayed upright, and they are also used in drones for flight stabilization. Other accelerometers are for example, able to detect gravitational waves.
- **Biometrics**[9]**:** biometrics authentication (or realistic authentication), refers to metrics related to human characteristics and it is used in computer science as a form of identification and access control. It is also used to identify individuals in groups that are under surveillance. Biometric identifiers are the distinctive, measurable characteristics used to label and describe individuals. Biometric identifiers are often categorized as physiological and behavioral characteristics. Physiological characteristics are related to the shape of the body (such as fingerprint, palm veins, face recognition, DNA, palm print, hand geometry, iris recognition, retina etc.), whereas behavioral characteristics are related to the pattern of behavior of a person, including typing rhythm, gait, and voice.
- **Capacitive**[10]**:** capacitive sensors detect anything that is conductive or has a dielectric different from that of air, such as sensors to detect and measure proximity, position or displacement, humidity, fluid level, and acceleration. Human interface devices based on capacitive sensors, such as trackpads, can replace the computer mouse. Digital audio players, mobile phones, and tablet computers use capacitive sensing touchscreens as input devices.
- **Current**[11]**:** a current sensor is a device that detects electric current (AC or DC) in a wire, and generates a signal proportional to it. It can be utilized either to display the measured current in an ammeter or can be stored for further analysis in a data acquisition system, or can be utilized for control purpose.
- **Flex/Force pressure sensors**[12]**:** flex sensors are sensors that change in resistance depending on the amount of bend on the sensor. They can be applied in a variety of sectors, such as robotics, bio-metrics, gaming gloves, in auto controls, in fitness products, measuring devices, assistive technology, musical instruments and joysticks.

[8] http://en.wikipedia.org/wiki/Accelerometer
[9] http://en.wikipedia.org/wiki/Biometrics
[10] http://en.wikipedia.org/wiki/Capacitive_sensing
[11] http://en.wikipedia.org/wiki/Current_sensor
[12] http://en.wikipedia.org/wiki/Pressure_sensor

- **Gyro[13]:** gyro sensors or gyroscopes, also known as angular velocity sensors, are devices that sense not only rotational motion but also changes in orientation. They are sensing devices that effectively augment human motion.
- **ID[14]:** ID sensors are defined as magnetic sensors, barcode sensors, touch ID sensors such finger recognition sensors, and RFID sensors.
- **Inertial Measurement Unit[15] (IMU):** an IMU is an electronic device that uses a combination of accelerometers and gyroscopes. IMU measures and reports a craft's velocity, orientation, gravitational forces and also magnetometers. IMUs are typically used to maneuver aircraft, including unmanned aerial vehicles (UAVs), and spacecraft, including satellites and landers. Moreover an IMU allows a GPS receiver to work when GPS-signals are unavailable, such as in tunnels, inside buildings, or in case of electronic interference presence.
- **Infrared[16] (IR):** IR sensors use infra-red light to sense objects in front of them and gauge their distance.
- **Imaging/Light[17]:** an image sensor is a device that converts an optical image into an electronic signal. It is used mostly in digital cameras, camera modules and other imaging devices. Furthermore, a LED can be used as a photodiode in light detection. This capability may be used in a variety of applications including ambient light detection and bidirectional communications. As a photodiode, a LED is sensitive to wavelengths equal to or shorter than the predominant wavelength it emits. A LED can be multiplexed in a circuit, such that it can be used for both light emission and sensing at different times.
- **Magnetometers[18]:** are measurement instruments used for two general purposes: (a) to measure the magnetization of a magnetic material, or to measure the strength and, (b) in some cases, the direction of the magnetic field at a point in space.
- **Proximity[19]:** they are proximity sensors able to detect the presence of nearby objects without any physical contact.
- **Radiation[20]:** radiation detector is a device used to detect, track, and/or identify high-energy particles, such as those produced by nuclear decay, cosmic radiation, or reactions in a particle accelerator. Modern detectors are also used not only as calorimeters to measure the energy of the detected radiation, but also to measure other attributes such as momentum, spin, charge etc. of the particles.
- **Sound[21]:** sound sensors can detect the sound strength of the environment by using as main component a microphone, while detecting sound waves.
- **Temperature[22]:** temperature sensors achieve quantitative measurements of temperature in order to accurately control a process, which process the signals they

[13] http://en.wikipedia.org/wiki/Gyroscope
[14] https://www.sparkfun.com/categories/144
[15] http://en.wikipedia.org/wiki/Inertial_measurement_unit#cite_note-1
[16] http://icrobotics.co.uk/wiki/index.php/IR_Sensors
[17] http://en.wikipedia.org/wiki/Image_sensor
[18] http://en.wikipedia.org/wiki/Magnetometer
[19] http://en.wikipedia.org/wiki/Proximity_sensor
[20] http://en.wikipedia.org/wiki/Particle_detector
[21] http://www.seeedstudio.com/wiki/Grove_-_Sound_Sensor
[22] https://controls.engin.umich.edu/wiki/index.php/TemperatureSensors

receive from sensors. They are applied in a variety of products, such as household ovens, refrigerators, and thermostats, in order maintain and control temperature and ensure that they function properly.

- **Weather**[23]**:** weather sensors are for example, humidity and temperature, altitude and pressure, barometric pressure or gas pressure sensors. (ref)

Moreover, a technology which is worth to describe further at this point is the Radio frequency identification (RFID) technology. According to [50], RFID is a wireless sensor technology which is based on the detection of electromagnetic signals. RFID systems comprise a read/write station and active (with own power source) or passive (power supplied by the read/write station) transponders (transmitter/responder), and can be used in a variety of applications. The range of possible applications is much greater, while RFID systems can be used for material tracking in manufacturing and logistics, for cash register applications in stores as an alternative to barcode scanning, or for localizing items or persons. RFID technology has been successfully applied in a number of scientific and technical fields, such as medicine, aeronautics industry, automotive industry, and retail industry.

The latest technology relies on individuals and items being tagged. In addition, the sensitivity of the tags can introduce challenges for the system. Traditional examples include protection against theft, access control, and billing. Specifically, if a Radio Frequency Identification (RFID) reader is positioned in a door frame to identify persons transitioning between rooms, the person can trigger the reader if they get close to the door, without necessarily moving to the next room. Otherwise, when motion sensors in combination with a RFID reader are placed on each side of the door, they are able to distinguish proximity from room transitions. Other sensing devices can be microphones (through the way of speaking and verbal explicit identification) and video cameras (through face recognition or explicit identification badges).

Moreover, according to [189], SMART DUST, a wireless network of tiny microelectromechanical sensors (MEMS), can monitor temperature, humidity and also inform for disasters, such as earthquakes, as well. It can be performed in the military as a remote sensor chip to track enemy movements and detect poisonous gas or radioactivity. Additionally, the biomimetic-fabric-based Sensing Glove, can be used to monitor hand posture and gesture and aims to conduct behavioral and functional studies [229]. Furthermore, the EasyLiving room at Microsoft Research [202] focuses on the collaboration of sensor techniques (e.g., IR motion sensors and electromagnetic field sensors and video cameras) and maintains an awareness of its occupants through computer vision, responds to voice and gesture commands, knows its own. Similarly, the Aware Home project of Georgia Institute of Technology [111] is a real house that consists of two floors including living rooms, bedrooms and baths, where ultrasonic sensors, radio frequency technology and cameras are used in order to observe users and test assistive technologies that should maintain their autonomy and independence of life.

[23] https://www.sparkfun.com/categories/152

2.2.2.2 Computer vision

The **Open Source Computer Vision** (OpenCV) is an open source computer vision library of programming functions with a strong focus on real-time applications. Such applications are image scan alignment, telemedicine [37], object analysis [52], security and intrusion detection systems [246], camera calibration [232], military applications [28], unmanned aerial, and ground or underwater vehicles [203]. OpenCV's application areas[24] in general include:

- 2D and 3D feature toolkits
- Egomotion estimation
- Facial recognition system
- Gesture recognition
- Human–Computer Interaction (HCI)
- Mobile robotics
- Motion understanding
- Object identification
- Segmentation and Recognition
- Stereopsis Stereo vision: depth perception from 2 cameras
- Structure from motion (SFM)
- Motion tracking
- Augmented reality

According to [106] and [130], OpenCV libraries, for instance, have been used in order to apply head and hand gesture recognition for Human-Computer Interaction. Similarly, researchers [97] have developed recognition methods of movable objects for the blind, based on the HuMoments method of Intel's OpenCV Library [96].

AForge.NET[25] , in addition, is a framework designed for developers and researchers in the fields of Computer Vision and Artificial Intelligence like image processing, neural networks, genetic algorithms, machine learning, robotics and so on. iSpy[26], based on AForge.NET, uses cameras and microphones aiming to detect and record movement or sound.

Another example based on Computer Vision techniques is the **reacTIVision**[27], a standalone application. ReacTIVision is an open source, cross-platform computer vision framework for the fast and robust tracking of fiducial markers attached onto physical objects, as well as for multi-touch finger tracking. It was mainly designed as a toolkit for the development of the **Reactable**[28] (see section 2.2.3), a table-based tangible user interfaces (TUI) and multi-touch interactive surfaces. Similar recognition techniques are also used by the Farm Game [131], which uses fiducial techniques in order small robots

[24] http://en.wikipedia.org/wiki/OpenCV
[25] http://www.aforgenet.com/
[26] http://www.ispyconnect.com/
[27] http://reactivision.sourceforge.net/
[28] http://www.reactable.com/products/

to be recognized in an augmented game environment [119] and also by **ARToolKit** used to develop a board game with a tangible interface [24].

Last but not least, additional applications based on Computer Vision implemented and developed from ICS-FORTH (AmI Program[29]) are the: a) AmI Playfield [162] that tracks the human position, b) the Augmented School Desk [10] applied for page recognition, iEat [132] implemented for dish and pen recognition, c) Interactive Documents [134] for book and touch recognition, d) Macrographia [82] for human body localization, and e) SESIL [133] for stylus recognizes.

2.2.3 Augmented artifacts

As technology becomes embedded in everyday objects people are provided with tools and processes in order to achieve more relaxing interactions with the surrounding environment. In general, an Ambient Intelligence environment can be considered to be an environment in which several Ubiquitous Computing (UbiComp) applications are hosted and make use of the infrastructure provided by the environment and the services by the objects therein.

Through clever use of advanced sensing and communication technologies, computers can be embedded into real world environment and, due to the continuous miniaturization of these technological components, even into smaller pieces. As the computer disappears, the objects in the surrounding environment become augmented with Information and Communication Technology components such as sensors, wireless communication modules, etc. These augmented objects also referred as "artifacts", play the major role in the merging of physical and digital space (i.e. tangible objects and physical environments are acquiring a digital representation). As reported in [61], Ambient Intelligence artifacts differ from traditional objects in a number of properties and abilities:

- **Information processing:** the information that an artifact processes can be descriptions of the context of use, data to be used for a task, guidelines on how to perform a new task (i.e. a program), messages to be sent or that have been received from other objects. The result of information processing is a set of services, that is, a set of abilities that appear in the digital space and relate to information; an artifact may offer or request services
- **Interaction with environment: a**rtifacts can perceive properties of their context of use (via their embedded sensors, or by communicating with other artifacts) and can also produce responses to these stimuli (via their actuators)
- **Autonomy: t**he operation of artifacts depends on electrical power; thus their autonomy depends on the availability of electrical power (which most of the times depends on the capacity of their battery)
- **Collaboration:** artifacts can exchange messages via (usually wireless) communication channels; the content of these messages may range from plain data to complex structures, including programs, database parts etc.

[29] http://www.ics.forth.gr/ami/

Artifacts possess two new affordances with respect to objects:

- **Composability**: artifacts can be used as building blocks of larger and more complex systems. This is a consequence of them possessing a communication unit and requires universal descriptions of tasks and services
- **Changeability**: artifacts that possess or have access to digital storage can change the digital services they offer. In other words, the tangible object can be partially disassociated from the artifact's digital services, as they are based on the manipulation of information.

RFIDice [90] is such an example of artifacts; a traditional six-side die (D6) that automatically detects the value of a dice throw (facing upwards at the end of a roll). In this case, the authors' concept idea was based on Radio Frequency Identification (RFID) technology able to offer countless new game options, while transferred to and played in the virtual world (i.e. a video or a computer game).

In particular and according to [90], the basic approach for using RFID to detect the value of a die was to add RFID tags to each inner side of a traditional D6 and embed an RFID antenna in the table surface. By detecting the tag that comes to rest on the antenna, the system is able to recognize the side that is facing down and can thus infer the side of the die that is currently facing up. This information can then be fed into the game system, where it might simply be displayed.

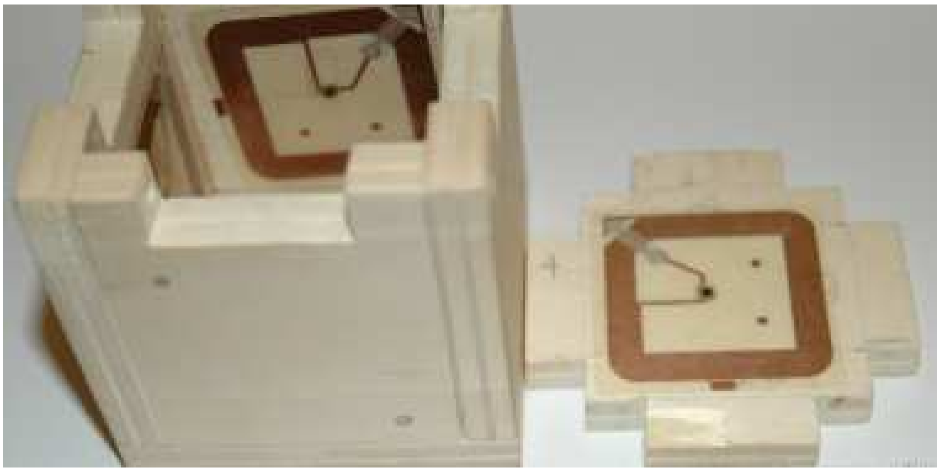


Figure 4: The normal dice with the lid open, revealing the enclosed RFID tags

RFID technology offers generally the following benefits:

- Low overall hardware costs (standard RFID equipment is continuously falling in price),
- No maintenance costs for dice (e.g., batteries, sensor calibration),
- Good integration into the environment (e.g. table),
- Small on-dice footprint (both size and weight),
- Easy dice manufacturing (many options for RFID tag packaging), and
- Little software costs (standard RFID software).

However, other similar industrial designed digital dice, named Dice+[30] uses Bluetooth technology to offer users a large range of facilities. Case by case, Dice+ encloses:

- **Accelerometer**: Dice+ includes accelerometer with magnetic field sensor that gives a three-dimensional orientation,
- **Magnetometer**: Dice+ detects moves that do not get handled by the accelerometer,
- **Temperature,**
- **Touch sensitive,**
- **Proximity**: a touch sensor identifies whether a side has been touched and how strong,
- **LEDs**: the device allows the customization of the light of each side, etc.

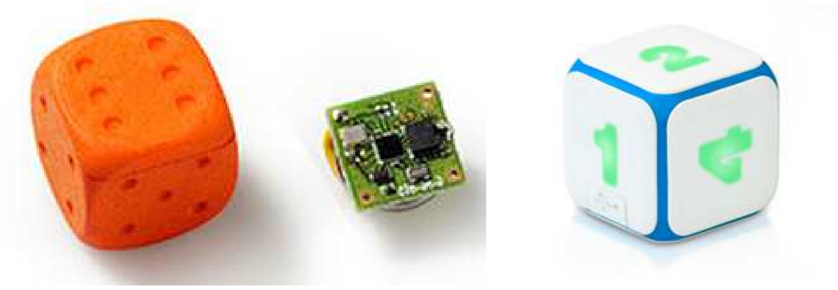


Figure 5: Samples of industrial designed digital dice

Figure 6: Example of a Dice+ and its' features

Furthermore, another smart object presented by Reactable Systems is the smart table **Reactable[31],** an electronic musical instrument that allows users to experiment with sound while enabling pedagogical, entertaining and creative possibilities. The intuitive and didactic aspects of the instrument, having the form of a table, allows the introduction to the musical basics and the creation of digital sounds. On the other hand, interactive tables can have a variety of use range such as educational purposes (further discussed in chapter 2.5.1) or even smart tables for restaurants[32]. The **iEat** [132] smart restaurant table aims at enhancing restaurant customers' experience in terms of entertainment, socialization, food selection and ordering. iEat provides interaction not only with users but also with physical objects placed upon its surface. User interaction is

[30] http://dicepl.us/
[31] http://www.reactable.com/products/
[32] http://itrestaurant.net/en/solutions/products#table

supported through moving or placing plates on the table, as well as through infrared light pens, which are stylus-shaped, LED-based, infrared light (IR) emitters.

Figure 7: Left is the Reactable and right a Restaurant Table

Figure 8: Smart tables with different use

Under the concept of how smart our environment is, according to [42], the MIT Things That Think (2006) group has developed intelligent devices such as smart hotpads that determine whether a pan is too hot to touch, a spoon that provides feedback about the temperature and viscosity of food, and a kettle that says how much longer you have to wait for tea. Additionally, Philips interactive tablecloth [168] weaves a power circuit into a washable linen tablecloth, so that devices can be charged when they are placed anywhere on the tablecloth.

Furthermore, the **Smart Sofa,** developed by scientists at Trinity College in Dublin, has many special functions that enhance user's experience when watching television, films or playing video games [188]. Similar projects identify individuals based on the weight and thus customize device settings around the house, they create an ultimate watching experience in action scenes by using vibrating, the sense of touch and the sense of smell, depending on the visuals and the sounds on the screen. Also, they automatically adjust in order to create the most comfortable and physically beneficial shape for the user's posture, by reading signals from the user's body.

Accordingly, the **Smart Chair** [38], extracts information about user activity from simple pressure sensors mounted under the legs of a chair. The smart chair detects different postures and activities such as working on a PC, watching a movie or eating in a continuous, real-life data stream. Another approach has been used to contact the area between the upper body and the chair to measure various physiological parameters. Examples are the measurement of vital signs in aircraft seats [201], car seats [231] and regular chairs [247].

Smart Tennis Sensor, a recently introduced product from Sony, helps tennis players to improve their skills and offer them a data knowledge while recording data during their play. The Smart Tennis Sensor is consisted of sensing technologies and vibration analysis mechanics that analyze a magnitude of player movements including shot count, ball impact spot, swing speed, ball speed and ball spin. Through highly sensitive wave and motion detection, the sensor can pick up multiple swing types such as topspin forehand, slice forehand and so on. Sony's Smart Tennis Sensor[33] wirelessly connects with smartphones and tablets using Bluetooth technology with performance data spontaneously visualized through a mobile application.

Figure 9: The Smart Tennis Sensor presented by Sony and its' mobile App.

2.2.4 Animated life-like characters in natural user interfaces

A growing number of research projects have begun to investigate the use of animated life-like characters in natural user interfaces because they present a priori a lot of advantages, which have also been validated by many authors. The authors of [12] describe interaction via an avatar and show that communication via an avatar can be useful. In detail, they propose a concept of Real-time Human Proxy for avatar-based interaction systems, which virtualizes a human in the real world in real-time, and which makes the virtualized human behave as if he/she were present.

In another research, an avatar is used as a personal assistant to interact with the television [227]. According to many authors, the use of animated life-like characters in

[33] http://blog.sony.com/press/sonys-new-smart-tennis-sensor-helps-to-elevate/

natural user interfaces presents a lot of advantages such as: a) social interaction [151], [154], [175], b) user attention[94], [115], c) naturalness, d) more information in the transmitted message [138], and e) trustworthiness and believability [118]. Research studies had explored remotely-controlled virtual humans for educational scenarios [160] as well as interactive virtual characters for presence in mixed reality [56], [160], [161] but not in a cross-platform, mobile, multi-device AmI environment.

2.3 Play and its contribution to child's development

The early years of a child are the time when his brain is developing, making connections and creating a network of skills that are built on throughout the entire life. Children are naturally inclined to create play situations and explore their environment. Through play they learn, practice and improve skills, involve in social roles and experience emotions; therefore, play is a significant dimension of early learning [182].

Play has a vital role in the everyday life of a child, as it helps enhance and develop fundamental skills, such as physical, social, emotional, intellectual, creative skills, etc. According to Piaget, play stimulates interest, initiative, experimentation, discovery, and imagination of a child in order to enhance his capacity to learn [172]. In particular, play can unite imagination and intellect in more than one way, while helping children discover things at their own pace and in their own way.

According to [109], play "paves the way for learning.", as it develops logical mathematical thinking, scientific reasoning, and cognitive problem solving. In addition, it fosters creativity and flexibility in thinking, since there is no right or wrong way to do things [220].

During play, children construct knowledge by combining their ideas, impressions, and intuitions, experiences and opinions. They create theories about their world and share them with others. They establish a culture and a social world with their peers, while they discover the intimacy and joy of friendship as they explore their own emerging identity. Due to the fact that play is self-directed, it leads to feelings of self-confidence and competence.

According to [107], young children learn the most important things not by being told but by constructing knowledge for themselves in interaction with the physical world and with other children – and the way they do this is by playing.

As children play, they learn to solve problems, to get along with others and to develop the fine and gross motor skills needed to grow and learn. Hospital staff, as an instance, often use play as a means of providing therapy to children. More specific, play helps a child to develop the following [22]:

- **Physical skills**: Gross motor skills are developed as a child learns to reach, grasp, crawl, run, climb and balance. Fine motor skills are developed as children handle small toys.

- **Cognitive concepts**: Children learn to solve problems through play (What does this do? Does this puzzle piece fit here?). They also learn colors, numbers, size and shapes. They have the ability to enhance their memory skills as well as their attention span and they move on to higher levels of thought as they play in a more stimulating environment.
- **Language skills**: Language develops as a child plays and interacts with others. This begins with parents playing games with their children and subsequently advances to more comprehensive levels such as telling stories and jokes.
- **Social skills**: Learning to cooperate, negotiate, take turns and play by the rules are all-important skills learned in early games. These skills grow as the child plays. As a result, children learn the roles and rules of society.

Through play, children recreate roles and situations that reflect their sociocultural world, where they learn how to subordinate desires to social rules, cooperate with others willingly, and engage in socially appropriate behavior. Over time, these competencies are transferred to children's everyday behaviors [65].

Researchers report that in today's primary schools instruction and test preparation have replaced art, music, physical education and play [79]. Many believe that play and academics are polar opposites and fundamentally incompatible. But a wealth of research demonstrates that play and academic learning are not incompatible. From dress-up to board games, from stacking blocks to art activities, research suggests that children's free-play fosters mathematics, language, early literacy, and social skills. It is also referred that adults can help children get the most out of their play by providing play materials such as books, pencils, paper, art materials and costumes, and by involvement in the form of questions, comments and suggestions [65]. Because play promotes brain growth and development, children who do not have sufficient opportunities to play will experience impaired brain development and flexibility [166].

According to [159], there is an optimum level of active social play necessary every day. Like sleep deprivation, play deprivation has adverse consequences. Without play, optimal learning, normal social functioning, self-control, and other cognitive functions may not mature properly.

Play is of extreme importance to human children, particularly during the 0-7 sensitive period [174]. There is little doubt that children deprived of play suffer considerable physical and psychological consequences, consequences which may be devastating to those affected. Children will adapt through their play to many changes in circumstances, like, for example the proliferation of computer toys, and may evolve new skills as a consequence. However, play deprivation is not about change but about an absence of those sensory inputs essential for the maintenance of humanness. Chronic play deprivation may have the effect of gradually dehumanizing the children it affects, with a consequent loss of their ability to care, to empathize and exercise compassion, or share the same reality as other children. The available evidence suggests that play deprived children become disturbed, aggressive and violent adults [174].

According to [89], there are many forms of play that evolve over the course of early childhood, variously described as exploratory play, object play, construction play, physical play (sensorimotor play, rough-and-tumble play), dramatic play (solitary pretense), socio-dramatic play (pretense with peers, also called pretend play, fantasy play, make-believe, or symbolic play), games with rules (fixed, predetermined rules) and games with invented rules (rules that are modifiable by the players). The following table describes each kind of play in conjunction with its age range of greatest incidence.

Kind of play	Description	Age range
Exploratory play/object play/sensory play	Very young children explore objects and environments (touching, mouthing, tossing, banging, squeezing). Sensory play appears in children's early attempts to feed themselves. As they get older, materials like play dough, clay, and paint add to sensory-play experiences.	0-2.5 years
Dramatic play (solitary pretense)	Many young children spend a lot of time engaged in imaginative play by themselves throughout the early childhood years. They invent scripts and play many roles simultaneously. Toys or props, (e.g., dolls, cars, action figures) usually support this kind of play. As children get older, they create entire worlds in solitary pretense, often with large collections of small objects or miniature figures.	3–8 years
Construction play	Children begin to build and construct with commercial toys (Lego, Tinker toys, blocks), with found and recycled materials (cardboard boxes, plastic tubing) and with a variety of modeling media (clay, play dough, plasticine). Older children play for extended periods with complex commercial model sets. Children across the age range engage in this kind of play by themselves and in groups, often combining it with episodes of solitary pretense or socio-dramatic play.	3–8 years
Physical play	Sensorimotor play begins as young infants discover they can make objects move; e.g., kicking the figures on a crib mobile or crawling after a rolling ball. Physical play in the preschool years often involves rough-and-tumble play, a unique form of social play most popular with little boys. Rough and tumble play describes a series of behaviors used by children in play fighting. Adults often mistake it as aggression. Older preschoolers engage in vigorous physical activity, testing the boundaries of their strength by running, climbing, sliding, and jumping, individually and in groups. This kind of play often develops spontaneously into games with invented rules.	3-8 years
Socio-dramatic play	Pretend play with peers - children take on social roles and invent increasingly complex narrative scripts, which they enact with friends in small groups.	3-6 years
Games with rules	Children begin to play formal games in social groups. These games have fixed and predetermined rules; e.g., card games, board games, soccer, and hockey.	5 years and up
Games with invented rules	Children begin to invent their own games and/or modify the rules of traditional playground games in their self-organized playgroups; e.g., tag, hide and seek, red rover, hopscotch.	5–8 years

Cooperative play	Children plan, assign roles and play together. Cooperative play is goal-oriented and children play in an organized manner toward a common end. Moreover, Cooperative play is a **"true social play"** in which children cooperate or assume reciprocal roles while pursing shared goals; play that is organized around a theme, with each child taking on a different role.	2 years and up

Table 1: Forms of play in childhood

In order to provide a measure of play ability, unified definitions have been created for four distinct categories of play and twelve levels aiming at sophistication spanning those categories [18]. The four categories used in the object play coding scale are exploratory, relational, functional, and symbolic play. Definitions of each play category are provided below and examples can be found in Table 2:

- **Exploratory play**: any child's action upon a single object that results from a visually guided reach and helps provide information about the object or environment. No functional relations exist between action and objects.
- **Relational play**: When two or more objects are used in combination with each other but are associated without regard to the functions or attributes of the objects.
- **Functional play**: Any conventional use of an object influenced by cultural properties of the object and simple pretend play actions.
- **Symbolic play**: Any scheme in a continuum of play schemes that incorporates items, attributes, contexts not actually present, or the substitution of objects.

Category	**Levels**	**Examples**
Exploratory	L1: indiscriminate actions	Grasp, rub, shake, bang, mouth
	L2: simple manipulation	Rolling toys, pushing a button
Relational	L3: takes combinations apart	Pull apart assembled toys, remove lids
	L4: general combinations	Stacking, scooping, pouring
Functional	L5: object directed	Covering with lids, dump payloads
	L6: self-directed	Imaginary drinking or talking on phone

Table 2: Elementary levels of object play along with canonical example

2.4 Knowledge models, assessment tools and young children developmental measures

There is a variety of assessment tools that professionals use in order to record issues involving functions and structures of the body, activity limitations and participation restrictions during the development of children (see APPENDIX IV - Common Tests and Measures of Development). This section analyses how ICF-CY [245] as a universal framework for measuring a child's development progress and Denver II [72], as Developmental Screening Test help define developmental disorders of young children.

2.4.1 A brief introduction

In 1973, the World Health Organization (WHO) commissioned Philip Wood to define a classification system, which resulted in the publication 'Impairments, Disabilities and Handicaps' [237]. This model, known as International Classification of Functioning, Disability and Health (ICF), has been used as a statistical, research and clinical tool in planning and designing social policy and for educational purposes [238]. Since 2001, ICF has been extended to address children and youth, in a version commonly known as ICF-CY (International Classification of Functioning, Disability and Health for Children and Youth).

ICF-CY provides a common and universal language for clinical, public health and research applications to facilitate the documentation and measurement of health and disability in children and youth. It is designed to record the characteristics of a child's development and the influence of its surrounding environment. ICF-CY can be used by providers, consumers and all those concerned with health (including occupational therapists), education, and well-being of children and youth [83]. The overall aim is to provide a unified language to describe healthy human functioning and health related states, outcomes and determinants, which can be used worldwide by clinicians, educators, policy-makers, family members, consumers and researchers.

The information provided by the ICF-CY, as mentioned before, may be used in a variety of ways including clinical, administrative, surveillance, policy or research applications. In each case, ICF-CY codes can be used to document a single problem or a profile of limitations defining a child's difficulties related to health and functioning. In this regard the ICF-CY can contribute to practice, policy and research in a number of ways. Specifically, the ICF-CY can:

- Provide a framework for inter-disciplinary practice
- Yield profiles of child functioning
- Clarify clinical diagnoses and co-morbidity
- Provide a functional basis for planning individualized treatments/interventions
- Offer codes for identifying intervention outcomes
- Provide the basis for documenting the gradient and hierarchy of change of functioning
- Standardize documentation of variables in research.

ICF-CY is divided into two parts [184]: a) Functioning and Disability, and b) Contextual Factors. These parts are further subdivided into components. Functioning and Disability contains two components: Body Systems (Function and Structure) and Activities/Participation. Contextual Factors also contain two components (Environmental and Personal factors). These components are defined as following with their corresponding functions (see Figure 10) [244], [245].

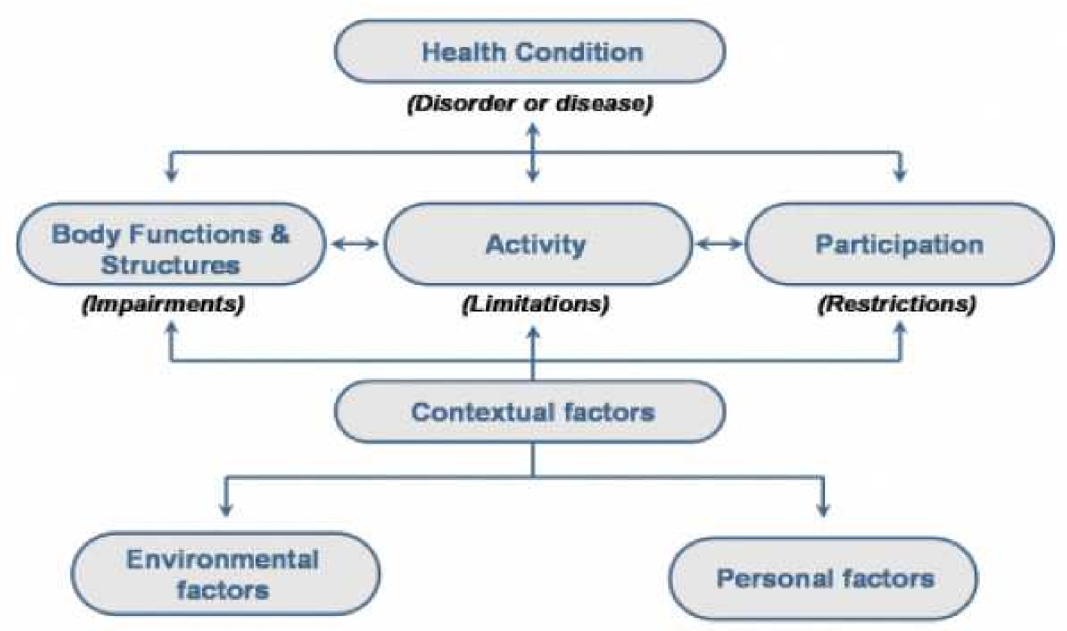


Figure 10: The World Health Organization's International Classification of Functioning, Disability and Health, version for Children and Youth (ICF-CY).

- **Body functions**: physiological functions of body systems
  - Mental Functions
  - Sensory Functions and Pain
  - Voice and Speech Functions
  - Functions of the Cardiovascular, Hematological, Immunological, and Respiratory Systems
  - Functions of the Digestive, Metabolic and Endocrine Systems
  - Genitourinary and Reproductive Functions
  - Neuromusculoskeletal and Movement-Related Functions
  - Functions of the Skin and Related Structures
- **Body structure:** Anatomical parts of the body, such as organs, limbs and their components
  - Structures of the Nervous System
  - The Eye, Ear and Related Structures
  - Structures Involved in Voice and Speech
  - Structures of the Cardiovascular, Immunological and Respiratory Systems
  - Genitourinary and Reproductive Functions
  - Structures Related to Movement
  - Skin and Related Structures
- **Activity and Participation:** The execution of a task or action by an individual and involvement in a life situation
  - Learning and Applying Knowledge
  - General Tasks and Demands
  - Communication
  - Mobility
  - Self-Care

  - Domestic Life
  - Interpersonal Interactions and Relationships
  - Major Life Areas
  - Community, Social and Civic Life
- **Environmental factors:** physical, social and attitudinal environments in which people live and conduct their lives
  - Products and Technology
  - Natural Environment and Human-Made Changes to Environment
  - Support and Relationships
  - Attitudes
  - Services, Systems and Policies
- **Personal factors:** the particular background of an individual's life and living, they comprise features of the individual that are not part of a health condition or health state.

2.4.2 ICF-CY in relation with occupational therapy

Occupational therapy (OT), is a science according to which health professionals provide support to a wide range of people with physical, psychological or developmental injuries or disabilities. They work with clients across the lifespan, from infancy to old age and have the common aim to promote, develop, restore and maintain abilities needed to cope with daily activities to prevent dysfunction and promote health [242].

Therefore, the development of the ICF-CY (WHO 2007) outlines its relationship to the changing paradigms of disability, explicating its components and proposing its application as a conceptual framework for clinical and educational assessment intervention in the field of Specific Learning Difficulties (SpLDs) in children. From a medical and psychological point of view [122], [208], SpLD as developmental disorder, occupies the specialists of OT while it is being involved in one or more of the basic psychological processes in understanding or using language (spoken or written). SpLD may manifest itself as an imperfect ability to listen, speak, read, think, write, spell, or do mathematical calculations. Such learning difficulties might be [127]:

- **Dyslexia:** difficulty with written language, particularly with reading and spelling
- **Dysorthographia:** disorder of learning characterized by an important and durable defect of assimilation of grammatical rules
- **Dyscalculia:** difficulty in learning or comprehending arithmetic and calculations
- **Dysgraphia:** deficiency in the ability to write.

The etiological framework for the classification, by diagnosis, of diseases, disorders and other health conditions, according to ICD-10 [243], is the following:

ICD 10	**Description**
Specific reading disorder (F81.0)	The main feature is a specific and significant impairment in the development of reading skills that is not solely accounted for by mental age, visual acuity problems, or inadequate schooling. Reading comprehension skill, reading word recognition, oral reading skill, and performance of tasks requiring reading may all

	be affected. Spelling difficulties are frequently associated with specific reading disorder and often remain into adolescence even after some progress in reading has been made. Specific developmental disorders of reading are commonly preceded by a history of disorders in speech or language development. Associated emotional and behavioral disturbances are common during the school age period.
Specific spelling disorder (F81.1)	The main feature in a specific and significant impairment in the development of spelling skills in the absence of a history of specific reading disorder, which is not solely accounted for by low mental age, visual acuity problems, or inadequate schooling. The ability to spell orally and to write out words correctly are both affected.
Specific disorder of arithmetical skills (F81.2)	Involves a specific impairment in arithmetical skills that is not solely explicable on the basis of general mental retardation or of inadequate schooling. The deficit concerns mastery of basic computational skills of addition, subtraction, multiplication, and division rather than of the more abstract mathematical skills involved in algebra, trigonometry, geometry, or calculus.
Mixed disorder of scholastic skills (F81.3)	An ill-defined residual category of disorders in which both arithmetical and reading or spelling skills are significantly impaired, but in which the disorder is not solely explicable in terms of general mental retardation or of inadequate schooling. It should be used for disorders meeting the criteria for both F81.2 and either F81.0 or F81.1.
Other disorders of scholastic skills (F81.8)	Developmental expressive writing disorder
Developmental disorder of scholastic skills, unspecified (F81.9)	Knowledge acquisition disability NOS (no other specified) Learning: -Disability NOS -Disorder NOS

Table 3: SpLD according to the ICD 10

The framework presented in this thesis employs a numeration of the developmental disorders of children according to ICF-CY's alphanumeric system, in which the letters "b", "s", "d" and "e" are used to denote Body Functions, Body Structures, Activities and Participation, and Environmental Factors. These characters are followed by a numeric code that starts with the classification number (1 digit), followed by the second level (2 digits), and the third and fourth levels (1 digit each). The categories are "nested" so that broader categories are defined to include more detailed sub-categories of the parent category. For instance, possible ICF-features characterizing a child with impairments are as shown in the following example:

- **A child with impairment in communication**
  - **d310.2** communicating with – receiving spoken messages
  - **d315.4** communicating with – receiving nonverbal messages
  - **d330.4** speaking
  - **d335.3** producing nonverbal messages
- **A child with problems of attention**

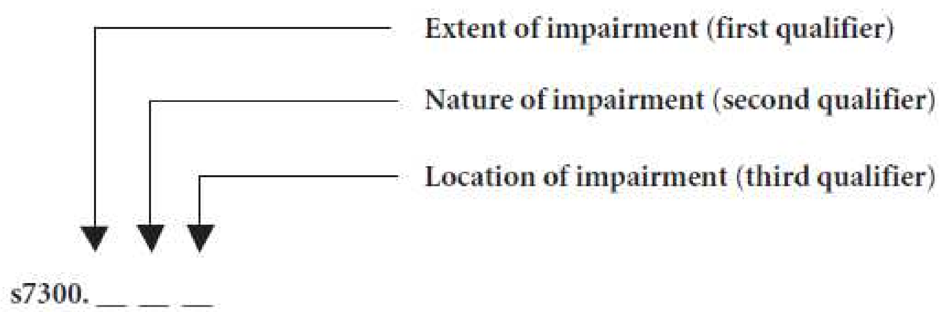
- **b1400.3** sustaining attention
- **b1402.4** dividing attention
- **d110.3** watching
- **d115.3** listening
- **d160.4** focusing attention

- **A child with problems of undertaking and completing task**
    - **b1641.3** organization and planning
    - **b1646.3** problem solving
    - **d2100.3** undertaking a single task
    - **d2102.4** undertaking a single task independently
    - **d2201.3** completing multiple tasks

Any individual may have a range of codes in each level. These may be independent or interrelated. The first qualifier for Body Functions and Structures, the Performance and Capacity qualifiers for Activities and Participation, and the first qualifier for Environmental Factors, as described below, all describe the extent of problems in the respective component.

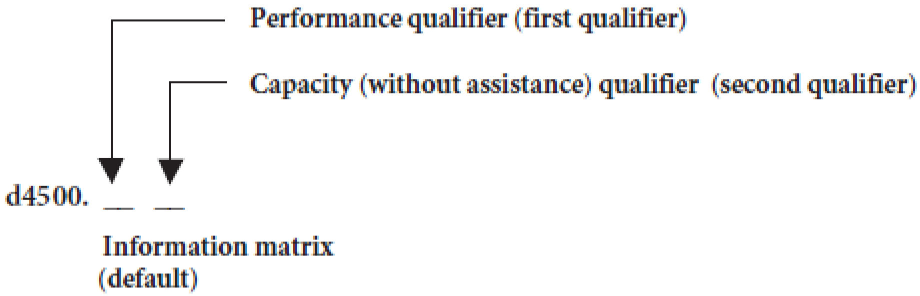
For the purposes of this research it is important to clarify how qualifiers are diversified for body structures. In particular, **body structures** are meant to refer to anatomical parts of the human body such as organs, limbs and their components, whereas **impairments** are problems in body function or structure as a significant deviation or loss.

Body structures are coded with three qualifiers. The first qualifier describes the extent or magnitude of the impairment, the second qualifier is used to indicate the nature of the change, and the third qualifier denotes the location of the impairment.



The descriptive schemes used for the three qualifiers are listed in Table 4.

Activities (the execution of a task or action by an individual) and **Participation** (as involvement in a life situation) is coded with two qualifiers: (a) the performance qualifier, which occupies the first digit position after the point, and (b) the capacity qualifier, which occupies the second digit position after the point. The code that identifies the category from the Activities and Participation list and the two qualifiers form the default information matrix [238].



According to WHO (2007) [245], the performance qualifier describes what an individual does in his or her current environment. Because the current environment brings in a societal context, performance as recorded by this qualifier can also be understood as "involvement in a life situation" or "the lived experience" of people in the actual context in which they live. This context includes the environmental factors – i.e., all aspects of the physical, social and attitudinal world. These features of the current environment can be coded using the Environmental Factors classification.

On the other hand, the capacity qualifier describes an individual's ability to execute a task or an action. This construct aims to indicate the highest probable level of functioning that a person may reach in a given domain at a given moment. To assess the full ability of the individual, one would need to have a "standardized" environment to neutralize the varying impact of different environments on the ability of the individual. This standardized environment may be: (a) an actual environment commonly used for capacity assessment in test settings; (b) in cases where this is not possible, an assumed environment which can be thought to have a uniform impact. This environment can be called the "uniform" or "standard" environment. Thus, the capacity construct reflects the environmentally adjusted ability of the individual.

Typically, the capacity qualifier without assistance is used in order to describe the individual's true ability which is not enhanced by an assistance device or personal assistance [243]. Since the performance qualifier addresses the individual's current environment, the presence of assistive devices or personal assistance or barriers can be directly observed. The nature of the facilitator or barrier can be described using the Environmental Factors classification.

All components (Body Functions and Structures, Activities and Participation, and Environmental Factors) are quantified using the same scale. Having a problem may mean an impairment, limitation, restriction or barrier depending on the construct. Appropriate qualifying words as shown in brackets below should be chosen according to the relevant classification domain (where xxx stands for the second-level domain number):

For this quantification to be used in a universal manner, assessment procedures have been developed through research [206]. Broad ranges of percentages are provided for those cases in which calibrated assessment instruments or other standards are available to quantify the impairment, capacity limitation, performance problem or barrier. WHO

(2007) illustrates the example, when "no problem" or "complete problem" are specified the coding has a margin of error of up to 5%. "Moderate problem" is defined as up to half of the time or half the scale of total difficulty. The percentages are to be calibrated in different domains with reference to relevant population standards as percentiles [245].

xxx.0	NO problem	(none, absent, negligible…)	0-4 %
xxx.1	MILD problem	(slight, low…)	5-24 %
xxx.2	MODERATE problem	(medium, fair...)	25-49 %
xxx.3	SEVERE problem	(high, extreme, …)	50-95 %
xxx.4	COMPLETE problem	(total…)	96-100 %
xxx.8	not specified		
xxx.9	not applicable		

Table 4: ICF-CY Qualifiers and Impairments

First qualifier Extend of impairment	**Second qualifier Nature of impairment**	**Third qualifier (suggested) Location of impairment**
0 NO impairment	0 no change in structure	0 more than one region
1 MILD impairment	1 total absence	1 right
2 MODERATE impairment	2 partial absence	2 left
3 SEVEE impairment	3 additional part	3 both sides
4 COMPLETE impairment	4 aberrant dimensions	4 front
8 not specified	5 discontinuity	5 back
9 not applicable	6 deviating position	6 proximal
	7 qualitative changes in structure, including accumulation of fluid	7 distal
	8 not specified	8 not specified
	9 not applicable	9 not applicable

Table 5: Scaling of qualifiers for body structures

As mentioned above, the knowledge of all components of ICF-CY is essential in order to collect and represent information about the development, as well as potential impairments or disabilities of a child. The appropriate information about the level of impairments at the level of body function (e.g., attentional problems, memory deficits, different components of mental language functions) can be used to help a child with a Specific Learning Difficulty (SpLD). Additionally, the difficulties that a child has experienced at the level of activity and participation (relationship with school, the capacity for reading, writing and calculating) and all the environmental factors that can be used for support (like family, friends, teachers, rehabilitators) and for rehabilitation (like specific technological instruments, all the services offered by the health system) are also essential information.

To summarize, ICF-CY is a valuable research tool because it combines dimensions of impairments at the body and body part level, person level activity limitations, and societal level restrictions of participation. ICF-CY provides an essential basis for the standardization of data concerning all aspects of human functioning and disability in the pediatric population by taking into account two relevant issues: (a) the dimensions of childhood disability which include health conditions, disorder, impairments, activity limitations as well as participation restrictions, and (b) the influence of the environment on the child's performance and functioning [183].

2.4.3 Assessment of young children's developmental skills

2.4.3.1 Typical child development

Development is usually considered as a process of growth and maturation that an individual undergoes throughout the life span. According to [191], child development is a progressive series of changes that occur in a predictable pattern as the result of interactions between biological and environmental factors. As mentioned in [98], there are ten principles of development that are still the subject of discussions in understanding the processes and issues in human development. According to those principles, development contains qualitative and quantitative changes, and is a product of intrinsic maturation and learning opportunities provided in the individual's environment. The human developmental pattern is predictable, has specific phases and predictable characteristics. Children follow a similar developmental pattern with one stage leading to the next, even though there are individual differences in the rate and the manner that they follow the pattern [98].

Theories of child development provide a basis in order to explore and understand the human factors involved in a child's play. Major theories of development and their basic assumption are presented below.

The basic assumption of the cognitive and developmental theory is that development is the result of the person's active participation in the developmental process in interaction with important environmental influences. According to [171], development is a discontinuous process characterized by abrupt changes from stage to stage. The theorist Lev Vygotsky [230] emphasized the role that culture and external influences play in leading the individual from one developmental level to the next. For Vygotsky, play is the source of development and creates the zone of proximal development (ZPD). Vygotsky's original conception of ZPD has been expanded, modified, and changed into new concepts such as "scaffolding", describing the support given during a learning process which is tailored to the needs of the student with the intention of helping the student achieve his/her learning goals [230]. Furthermore, ZPD and scaffolding can be used in various learning contexts where task modeling, gradient and guidance/coaching are needed.

Behavioural theory adopts that development is a function of the laws of learning and the environment has important influences on growth and development. Its main impact is in the area of systematic analysis and treatment of behaviour [98].

Psychodynamic theory refers to the individual differences as well as the normal growth as resulting from the resolution of conflicts of human beings [98]. In the psychosocial model, development results from the interaction between biological needs and social demands [98].

Biological and maturation theory is based on the acceptance that biological factors and the evolutionary history of the species determine the sequence and the content of development [98]. The Gesell Developmental Schedules [17] is an instrument of the status of a child's motor and language development and personal-social and adaptive behaviours. The Denver Developmental Screening Test (DDST) for children 0-6 years old is one of the widely used assessment tools that have been established for children based on based on Gesell's initial schedules [70].

2.4.3.2 Development of child's play skills

According to [204], child's play develops in several stages from passive observation to cooperative purposeful activity. There are many ways of categorizing play behaviors and skills. However, there are some common elements that can be used to distinguish play from other types of activity, such as:

- play is entertaining and enjoyable
- it is energetic, incentive and self-chosen activity
- allows the player to create or/and modify the play situation
- involves pretending and imagining
- the outcome is less important than the process
- play relatively has no rules; games have rules.

Children are naturally inclined to create play situations and explore their environments. Through play they learn, practice and improve skills, involve in social roles and experience emotions. Therefore, play is the most common therapeutic and educational intervention for children.

According to [204], Table 6 illustrates typical play behaviors and skills of play development in the age range from two to seven years old.

Age	Typical Play Behaviors and Skills
2-3 years	With the increase in use of language during this period, the child engages in symbolic and pretend play and begins to shift from parallel play to more interactive forms of play; talks to self during play and begins to use language when playing with others; shows a variety of emotions during play and likes to role-play adult roles; may enjoy action figures, dolls and other pretend people; may continue to be possessive of toys; likes to imitate, gross motor play incudes using playground equipment with some assistance, learning to ride a tricycle, jumping with both feet clearing the ground together, simple ball play (e.g. kicking and tossing a medium sized ball) and running around, climbing and dancing; fine motor play includes painting and scribbling; large construction toys and insert puzzles and more complex cause – effect toys that introduce preschool concepts such as colors, shapes, letters, and numbers; continues to be interesting in picture books, enjoys

	sensory play like Play Dough (clay), water and sand play.
3-5 years	Engages in creative and group play, and associative play dominates by the 4 year of age as the child learns to share and take turns and is interested in friends; continues to enjoy role-playing and dressing up, and creating elaborated pretend play situations; may begin to play simple board games, such as checkers or Candyland[34] ; with respect to gross motor play, the child becomes proficient in playground equipment, including being able to pump a playground swing; likes to ride a bike with training wheels; may begin to participate in more structured recreational activities, such as swimming, dance, and skiing; enjoys running around, jumping, hopping, climbing, and ball playing; manipulative play skills include painting and coloring, simple drawing, coping basic shapes and same letters, scissors use and simple craft activities, construction toys and computer play; begins to develop an interest in the finished product of construction play; may became more interested in television and may begin to play video games.
5-7 years	Enjoys games with rules, such as board games, and becomes much more involved in organized sports and recreation in the community, learns specific skills such as swimming, skating and bike riding or playing a musical instrument, and preferences for certain play activities became more prevalent; plays well with others and enjoys social interaction and play to reach a common goal, understands concepts of cooperation and competition, and the importance of friendship increases; independence during play increases with the extensions with neighborhoods and the homes of peers; secondary play and leisure activities (watching TV, reading, and playing computer games or video games) may also increase.

Table 6: Typical Play Behaviors and Skills

Many standardized assessment tools have been designed to measure various aspects of performance and skills in childhood, such as:

- developmental evaluation and screening tools
- tools for measuring functionality in specific activities of child occupations (day living activities, play, education etc.)
- assessment of specific skills (motor, sensory, perceptual, etc.)
- assessments for psychological functioning
- tools for environmental control.

2.4.3.3 Standardized assessment tools

Development tests are tools that are used to help therapists measure a child's developmental progress from infancy through adolescence. They may help to indicate early signs of a developmental problem and discriminate normal variations in development among children, depending on the age of the child. Such assessment tools are designed according to the expected skills of children at a specific age. The types of developmental assessment principles include:

[34] Candyland is a simple racing board game which requires no reading and/or minimal counting skills.

- developmental screening to identify children with special needs, developmental delays or school difficulties
- diagnostic evaluation, depending on the screening, to confirm the presence and extent of a disability
- readiness tests to assess a child's specific skills
- observational and performance assessments to provide ongoing information about a child's development.

Types of development tests are [116]:

- infant development scales
- sensory-motor tests
- speech and hearing tests
- preschool psycho-educational batteries
- tests of play behavior
- social skills and social acceptance tests.

There are many scales commonly used to evaluate and measure developmental skills, such as Peabody Test [44], Denver Developmental Screening Test [68] and Millani Comparetti scale [217] etc. (see APPENDIX IV - Common Tests and Measures of Development). In the current research work, the Denver Scale II has been used as evaluation tool.

2.4.3.3.1 Denver Developmental Screening Test (DDST)

The Denver Developmental Screening Test (DDST), commonly known as the Denver Scale, was originally designed at the University of Colorado Medical Center, Denver USA. The Denver Scale aims at screening cognitive and behavioral problems in preschool children [69]. More specifically, it enables the professionals to identify children whose development deviates significantly from that of other children of the same age, while warranting further investigation to determine if there exists a problem requiring treatment.

The revision and update of DDST, Denver II[35], is a widely accepted pediatric screening tool for ages up to 6 years old frequently used for testing the domains of: (a) personal social (such as smiling), (b) fine motor adaptive (such as grasping and drawing), (c) language (such as combining words), and (d) gross motor (such as walking). The scale reflects the percentage of a certain age group able to perform a certain task (e.g. 90% of three-month-olds smiles spontaneously, 90% of 21-month-olds speaks three words other than "mom" and "dad", 90% of 5-year-olds hops on one leg, etc.).

There are five unique features of the test that generally differentiates it from most other developmental screening tests [147]:

- Its validity rests upon its meticulous and careful standardization reflecting the US 1980 census population [72]. Most other developmental screening tests base their

[35] http://en.wikipedia.org/wiki/Denver_Developmental_Screening_Test

validity on measures of sensitivity and specificity. Most such studies suffer from one or more of the following: small sample size, verification bias, inappropriate/non-equivalent test bias, procedural bias, spectrum bias and incomplete reporting of results [34], [20], [9].

- Since the test depicts in graphic form the ages at which 25%, 50%, 75% and 90% of children performed each item, it enables the examiner to visualize at any age from birth to six years how a given child's development compares with that of other children.
- The test has separate norms for subgroups of the population based on sex, ethnicity and maternal education when the subgroups differed by a clinically significant amount from the total group or composite norms.
- The test is primarily based upon an examiner's actual observation rather than parental report.
- It is ideal for visualizing on one page the developmental progress of children whether or not their development is being monitored for optimal child care or because the child's development is of special concern.

The Denver II Test includes 125 items in total, which are recorded through direct observations of the child [77]. For some points, parents report whether their child is capable of performing a given task or not. As shown in APPENDIX V - Denver Developmental Screening Test), the data are presented as age norms, similar to a growth curve. The more items a child fails to perform (passed by 90% of his peers), the more likely the child manifests a significant developmental deviation that warrants further evaluation [71].

2.5 Children in AmI environments

Augmented artifacts as well as computer games are broadly considered as emerging technologies offering a high potential to foster and support learning [176] in AmI environments for young children. One of the key issues to be investigated to nourish this general claim is the relationship between academic performance and the use of computer games in relation to different disciplines, educational settings and underpinning cognitive tasks. Currently, few examples of augmented artifacts applied while playing exist, however the relationship between games, learning objectives and learning environments has not yet been explored (i.e., which kind of games better serve the scope of fulfilling specific learning objectives and how this can be done in specific learning contexts) [137] [145].

2.5.1 Gaming technologies in AmI environments

2.5.1.1 Augmented artifacts for play and play environments

There are strong cognitive-developmental reasons why interacting with task-appropriate physical objects is the best learning environment for young children. Embedding interactivity into physical objects allows supporting traditional exploratory play with physical objects that can be extended and enhanced by the interactive power of digital technology [181]. Piaget and developmental psychologists [171] have emphasized the critical importance of manipulation of physical objects for young children's cognitive development. In addition, also emphasized is the importance of play in facilitating a child's development [230]. The objects used in children's play can include anything that can be tangible, such as sticks, rocks or cardboard boxes, and often include especially designed toys such as dolls, miniature cars and trucks, blocks, and so on.

Smart artifacts that support young children interacting in order to play and in the meanwhile enhance their learning skills include I-Blocks, the Cube to learn and other interactive games such as puzzles. I-Blocks, for example, allow children develop their mathematical skills through arithmetic calculations that they must carry out in order to build an interactive construction of blocks. On the other hand, Cube to learn, is a user interface toy that encourages children to develop and enhance a variety of skills while answering quiz questions. Additionally smart Jigsaw Puzzle and the Farm Game are cases of games using technology methods to entertain and at the same time to educate children.

I-Blocks are physical artifacts to support not only learning by construction, but also programming by building [126]. By attaching a number of basic building blocks, each containing a microprocessor and communication channels, together, the user constructs an artifact that can perceive input, process the input data, and produce output (see Figure 12). I-Blocks can be manipulated to create both physical functional and conceptual structure in an easy and fast manner. According to the authors, I-Blocks support their claim that both body (physical structure) and brain (functional structure) play a crucial role in intelligence. The body and the brain of natural existences co-evolve to fit to each other and the surrounding environment.

Figure 11: I-Blocks

Each I-Block has a physical expression (e.g., a cube or a sphere). When attaching more I-Blocks together, a user may create a physical structure of I-Blocks that process and communicate with each other, depending on how the I-Blocks are physically connected to each other. Interaction with the surrounding environment happens through I-Blocks that obtain sensory input or produces actuation output. So the overall behavior of an 'intelligent artifact' created by the user with the I-Blocks depends on the physical shape of the creation, the processing in the I-Blocks, and the interaction between the creation and the user or the surrounding environment.

The form of the blocks is a Lego Duplo brick. I-Blocks (see Figure 12) allow users to build a program by connecting blocks of the following three types:

- Input blocks with a sensor or mechanism to set a value
- Output blocks to produce, for instance, a tone, light or signal and to display a value
- Operator blocks for arithmetical or logical operations.

Interesting to mention is that there is also an interface that allows students to program I-Blocks for different functions in an easy way. The QEL Micropro program is used to download programs to the microchip inside each I-Block.

Figure 12: Arithmetic I-Blocks that performs an arithmetic calculation (left), The construction turns towards light, changes LED activation, and makes random moves, depending on the interaction with the environment (right)

Cube to learn is a tangible user interface for the design of a learning appliance [225]. Using the common shape of a cube, it provides a general learning platform that supports test-based quizzes where questions and answers can be text or image based. Exploiting the physical affordances of the cube and augmenting it with embedded sensors, it presents different learning appliances as playful learning interfaces for children.

Learning Cube is a digitally augmented physical cube. It is enriched with a display on each of the six faces and a speaker inside. The display and the speaker are controlled by an embedded hardware platform.

In Figure 13, in the left part of the image, the opened cube with the hardware in the base and wires going to the display on the opened side of the cube are shown. In the right part of the image the closed cube, with the vocabulary trainer appliance running, is shown. The cube is a good platform for quizzes and tests. The affordance of the device is known to every potential user of all age-groups, from young children to adults. The device can be picked up, rotated and played with, thrown and shaken, and put down again. Additionally, relying on the experience with dice games, people expect to find different information on each side of the cube. This information can be textual or figurative.

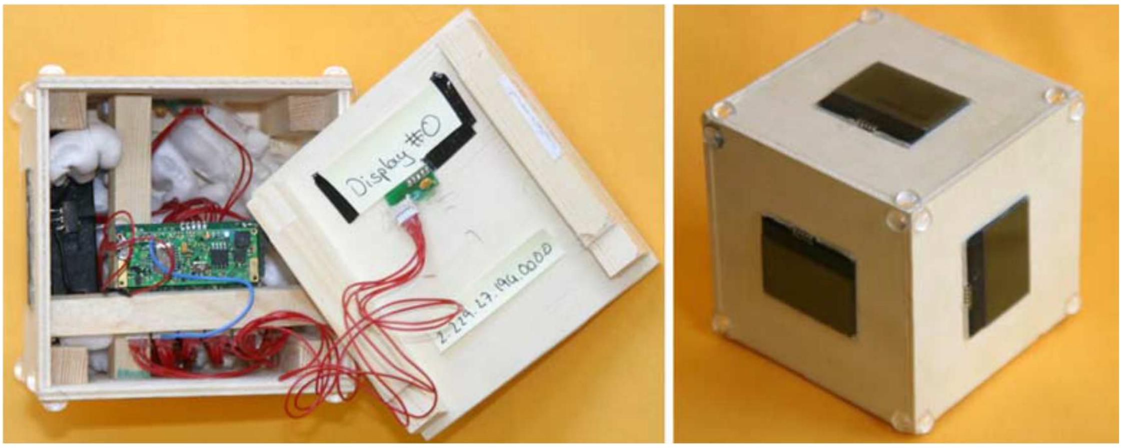


Figure 13: The Learning Cube. The cube is built around a microcontroller platform and has a display on each side. Acceleration sensors allow the detection of orientation and movement

The basic platform offers the function of a multiple choice test system. In each step one question and five answers are used. One of the answers is correct, the others four are wrong. The question is shown on the top display and the answers are randomly distributed to the other displays. The interaction operated by the user is to turn the cube to the side with the right answer and then shake it. If the right site is selected, the program moves on to the next question; if not, the user can try again.

Figure 14: A small child playing with the Learning Cube. Even children as young as 3 years could instantly use the appliance

As an experiment, an application for children was developed, where letters had to be matched. On the top, a letter was displayed (e.g., A) and different letters were displayed on all other sides including another A. The objective of the game was to find the matching letter and then select it. The game was designed for children starting to learn letters. The main difficulty is that the child had to find the letter even if it is upside down.

Figure 14 shows a 3-year-old girl playing with the appliance. This first experiment showed that the game was challenging for young children; features such as shaking and turning the cube around appeared pretty difficult to execute, kept the game interesting enough for a certain time and engaged children in constantly trying out gestures in order to find out the solution. Although, in some tests, there was a low recognition rate for the shaking gesture, children were still interested in playing, while trying to make gestures in a way that the system could recognized them.

Smart Puzzle, a two-dimensional Puzzle, enhances the ability of students to solve mathematical and scientific puzzles [195]. It is developed based on techniques implemented in a Tangram[36] (i.e. a dissection puzzle), with a smart-objects interface, and provides multimedia feedback through a computer application as children play with the puzzle pieces. In detail, when students first approach the puzzle, they are greeted by an opening screen that explain the objective and the rules of the puzzle. If a student clicks the button, an animation figure provides voice-over that explanation. This explanation then remains available to the students throughout the game. The interface, moreover, gives students a hint when they appear not being able to move on or when they are not making progress toward the solution. Again, students must click a button to actually see the hint.

The author supports that by showing children the physical puzzles behind math and science, the puzzle can gain their interest and help them to understand and appreciate more abstract concepts. Puzzles equipped with smart object interfaces can passively “observe” students as they play, offering help, hints, reminders and explanations only when they are needed.

Computer vision techniques have been used in order to track the smart objects [195]. The authors installed a color QuickCam with an adjacent light source to capture images of the current puzzle arrangement on a table. A tinted reflective surface on the puzzle pieces helped to identify the individual pieces, and made it easy to threshold out everything else in the scene. The tracking module used standard segmentation and feature extraction techniques to find individual puzzle pieces in the image. By comparing images over time, it is able to focus on those areas where the puzzle arrangement has changed. The tracking module returns information about the position and orientation of puzzle pieces.

Another similar project, the **Smart Jigsaw Puzzle Assistant** [27], uses miniature RFID tags and a RFID scanner, giving to each puzzle piece its unique ID. While tagging all the jigsaw puzzle pieces with this technology, the author succeeds to avoid the problem of not going further if the puzzle is pretty complicated and the player cannot move on.

The Smart Jigsaw Puzzle Assistant (SJPA) according to [27], enables the user to efficiently seek out further matching pieces of a jigsaw puzzle by simply using a small RFID antenna as a detector. The user moves the detector over the remaining jigsaw

[36] http://en.wikipedia.org/wiki/Tangram

pieces until it detects the specific piece that can be added to the already combined pieces of the puzzle.

Figure 15: Left: A matching piece with an attached miniature RFID tag. Right: RFID antenna and below the corresponding μ-chip reader device by Hitachi

A very interesting approach to augmented artifacts in children entertainment and education is the **Farm Game**, a tangible tabletop application for children [131]. The main objective of the Farm Game is to contribute enhancing both motor-skills and cognitive development. The game consists of a virtual farm with animals and a farmer as a virtual agent, represented on a 3D virtual environment that is shown on the monitor (see Figure 16). The farm animals are plastic animal toys with fiducials attached on their bases tracked on the tabletop. Through the Farm Game children can play freely with the animals all over the meadow (tabletop). The animals are represented on the screen as animated characters and they can walk and make sounds while children drag them around the tabletop. While playing the Farm Game, children are required to feed the animals by dragging the real toys through the tabletop to the right position, to take care of them and their surrounding and also to look after baby animals by keeping them near to their mother. The virtual agent, apart from giving instructions to users how they should proceed in the game, also gives them information and explains the role of each character in the game.

As the authors report [131], the game environment is implemented using Maxine software [16], which is a 3D engine for the management of virtual environments and virtual characters in real time. The authors also mention that through Maxine it is possible to load geometrical models, animations, textures, videos or sounds as they are required in the virtual representation (the communication used between the image recognition software and Maxine is made via TCP-IP). Even though Maxine is a very generic engine, it has been specially oriented to work with virtual characters. It implements body and facial animation, lip-synchronization, and allows the character to show emotions through facial expressions, answers and voice. Maxine agents can act as virtual tutors or virtual companions that help children in their learning process, proposing new activities depending on the children characteristics and evolution.

Figure 16: Illustration of the tangible tabletop prototype

Figure 17: Wooden toys with fiducials attached to their base

The Farm Game application is based on ReacTIVision software [21], a recognition software which is able to identify each toy as it is placed on the table. ReacTIVision uses a standard web-cam, retrieving position, orientation and the ID number of any fiducials that are within view of the camera and after tracking the fiducials then sends the data to the game through TCP-IP packets.

2.5.1.2 Ambient edutainment systems

Technologies and learning environments for children have moved beyond the computer interface into tangibles, mixed-reality, virtual reality, ubiquitous computing, and embedded interactions. Educational AmI environments aim at providing education through real world context activities and are characterized by their facility to suit non-skilled learners and educators for any type of learning or teaching [162]. In this context, learning activities are claimed to be more effective when involving the learners' active engagement in groups and intensive interpersonal interaction in real world contexts [186].

An interesting approach of edutainment systems for children in AmI environments is through storytelling. FaTe [75] as an example of edutainment environment for young children, allows children to play, communicate, explore, build their own stories and create their own narrative flows in a collaborative environment. Similarly, Kidpad [103] is a collaborative storytelling tool for children, which allows them to create their own hyperlinked story scenes and link them together in a two dimensional space. Some other researchers have exploited the fun of virtual reality (VR) in building their edutainment environments for children, such as the storytelling Holodeck project. According to [36], Holodeck consists of a number of virtual actors (VAs) in the same environment. The interaction with the VAs take place through natural language or speech and the way the user can participate in story generation is either physically, verbally or interjectionally through off-stage intervention. Along the same lines, Story Toy [67] lets the children play with linear and branched narrations on a toy farm with electronic sensors making use of voice but not of images, whereas SIDE Project [173] aims to improve the social skills of adolescents with Asperger's Syndrome.

AmI Playfield [163] is an application that provides technological support in an AmI edutainment environment, encourages playful learning and learning by participation while providing the basis for natural (kinesthetic) and collaborative interaction. More specifically, the players must gather the apples available on the floor and on the meanwhile they calculate simple arithmetic operations required for moving from one place to another. Educational tabletop mini-games which combine learning, entertainment and ambient intelligence include: (a) multiple choice quiz game and (b) geography-related game [120]. Both games engage users – children in a collaboratively activity where they learn through experience, either by selecting the best possible answer or by memorizing the location of cities or historical monuments, while interacting in a physical manner with user interface platforms in an AmI environment.

The interactive wall game Paximadaki [81] is an interactive game which involves physical activity as a means of interacting with the game while using Kinect's depth camera in order to render a virtual shadow of the players[37]. Players must use their shadows to put the falling numbers which are required (e.g., only odd number) into their basket, until a certain amount of numbers have fallen. In case of sending the wrong number into the basket, a point will be removed from the overall score.

Last but not least, Smart Kindergarten is a system that allows children to explore and interact with objects, such as toys, in their environment. Researchers support that the experience of having the environmental respond (casually) to children's actions is one key of their development [196]. This system enhances the education process by

[37] http://ics.forth.gr/ami/projects/view/All/Paximadaki

providing a childhood learning environment that is individualized to each child and allows unobtrusive evaluation of the learning process by the teacher.

2.5.2 AmI technologies embedded in assessment tools used from OT to monitor progress and developmental issues

It is widely recognized that the role of Information and Communication Technology (ICT) in the future of healthcare will be significant and the healthcare professionals will need to be computer literate in order to function effectively in an increasingly digital environment [152]. The World Health Organization (WHO) has discussed the benefits of ICT within the healthcare sector and state that it has the potential to improve the exchange of data leading to better access to information and provides learning tools for healthcare professionals, students, patients and the community as a whole [105]. Moreover, WHO supports that the use of ICT within the field of healthcare is becoming an increasingly important aspect of clinicians professional practice, improving the delivery of health services and communication between healthcare workers, as well as enhancing the decision making process through the efficient flow of information.

Technological advancements continually influence occupational therapy practice methods and create new tools for intervention. One of the areas of advancements in healthcare sector is virtual rehabilitation, where technological changes have driven to innovative and enhanced treatment methods in occupational therapy. More specific, virtual rehabilitation is the use of virtual reality (VR) and virtual environments (VE) in rehabilitation. Virtual rehabilitation has received increasing attention from researchers and clinicians who recognize potential therapeutic benefits due to the immersive nature of the medium [93].

This section discusses how ICT effects Occupational-Therapy (OT) technics regarding monitoring progress and identifying developmental issues of a child such as language skills, and behavior analysis.

2.5.2.1 Occupational therapy while playing

Nowadays, there is a large range of software solutions that promote the development of a child and monitor potential developmental issues, as well. These technologies include programs that focus on speech sounds, phonological awareness, vocabulary development, sentence comprehension and story-telling. Many programs are supported by text, others by sign language and some by both [148].

Developmental skills, according to [236] can range from banging a toy on a table to displaying socially appropriate expressions. Specialists in OT often use a subset of these milestones in screening diagnostics, and recent research in psychology suggests that the observation of object play interactions may help identify early indicators of certain developmental delays [4],[18]. Psychologists have created a coding scheme which quantifies the levels of sophistication displayed by infants while engaged in object play, as play is the most common therapeutic and educational intervention for children. Researchers used this scheme to identify the highest level of sophistication reached per child and found that the duration of play at each level differed between typically developing children and children with autism [18].

Researchers have explored technologies to assist individuals with developmental disorders such as autism. Most of these applications have been developed for use by the users themselves rather than their Occupational Therapists. Such a technology includes Simone Says, a system that uses voice recognition technology to teach and analyze language skills [121]. Moreover, other existing work encourages vocalizations for children with autism by using visualizations [84]. The Discrete Trial Trainer (The Discrete Trial Trainer - DTT, 2004) is a commercial software product that attempts to replace the therapist in Discrete Trial Training therapy by administering similar therapy and education through the computer. While Simone Says and Discrete Trial Trainer are both used by the individuals themselves, according to [112], they can ease some of the burden on therapists by allowing a computer to administer therapy, thus leaving the therapist more time to handle other aspects of care.

According to a study, the Child's Play [236] system supports a subset of play activities, while automatically generates quantitative data from observations of children's behavior, based on the coding scheme of related work [18]. These measures include factors like: (a) the frequency with which an object is played, (b) the time spent attending between different objects, and (c) the highest level of play sophistication reached by a child. Child's Play uses statistical pattern recognition techniques of sensor augmented toys and a mobile computing platform in order to receive data and identify play activities associated with developmental skills through the way in which children interact with objects [235], [236].

Figure 18: The Child's Play system selects rich data from an infant shaking a toy and uses statistical methods to annotate the date for late review

Figure 19: A child with speech disabilities uses an iPad during OT

The Children's Center partners with Oklahoma ABLE Tech[38]activated the Assistive Technology Act Program in order to provide services for children with complex medical and physical disabilities, such as a five year old boy who is diagnosed with apraxia of speech (see Figure 19). The Children's Center recommended to use an iPad during his

[38] http://www.ok.gov/abletech/Assistive_Technology/SuccessStory.html

therapy. In this way, the boy learned how to utilize the communication device and increased his ability to express his wants and needs. This level of communication, in turn, helped him to decrease his behavioral outbursts and also gave to his therapist the opportunity to monitor and enhance his behavior skills.

Other technologies have focused on how to teach children social skills and aid in communication. Research has explored storytelling with virtual peers to teach social skills to children with autism in a more comfortable setting [223], while others at MIT have looked at how images of people can be emphasized to help understand subtle emotional cues [108]. Related work has explored how a tabletop, multiplayer game, called SIDES, can be used to encourage children with autism to learn social interactions and turn-taking [173]. These types of social applications can be used in conjunction with therapists to provide a more rich education in social skills.

Furthermore, Plush Cube [18], a system based on micro sensors, is useful for research made by occupational therapists in order to extract information about the developmental skills of a child. Taking into consideration that the way in which infants play with objects can serve as an early indicator for developmental delays, Plush Cube, which detects when a toy has been touched and how tightly it has been grasped, can be a considerably important tool for the professionals.

Last but not least, CareLog helps occupational therapists collect better data for decision-making [86]. CareLog seeks to support teachers in a classroom in order to diagnose the causes of children’s behavior by allowing retroactive video capture of events (through selective archiving [87] to help support systematic decision-making on the cause of the behavior.

By facilitating therapy in a controlled virtual environment, occupational therapists are able to offer functionally relevant and ecologically valid therapy and assessment [185]. Ecological validity refers to how performance in an experimental context relates to and is predictive of behavior in the real world. In addition to immersion, there has been increased interest in VR due to its motivational nature; according to [85], individuals using VR tend to have fun and are thus more motivated to continue therapy [23].

An example of playing as a method of occupational therapy is the Wii game console that uses interaction techniques in order challenge the users by improving their body functional skills. Wii is a video game system that relies on three dimensional movements to cue real-time responses within the software (see Figure 20) released from Nintendo, November 2006 [41]. Wii game Console, apart from its’ entertainment use, has also therapeutic character, as professionals apply it in Occupational Therapy, as well. It has been observed that occupational therapists begun to use Wii games as a part of their regular treatment at the Glenrose Rehabilitation Hospital [85], a tertiary rehabilitation center in Edmonton, Alberta. Applying the principles of activity analysis, therapists use the Wii system and the Wii Sports software as a functional therapy task not only for adults but also for young children. The patients are oriented to the system and closely monitored by the therapist throughout the session. Wii Sports, for instance, include five different activities: tennis, baseball, bowling, golf and boxing and each of them requires from the player to respond with specific movements in order to play the game. It has been observed [85], that after using such technology methods during OT, clients stay in therapy session longer than usual, engaging in social interaction and meaningful occupation. Moreover, occupational therapists report that they can continue to work on identified client-centered goals while using video games as a therapy method.

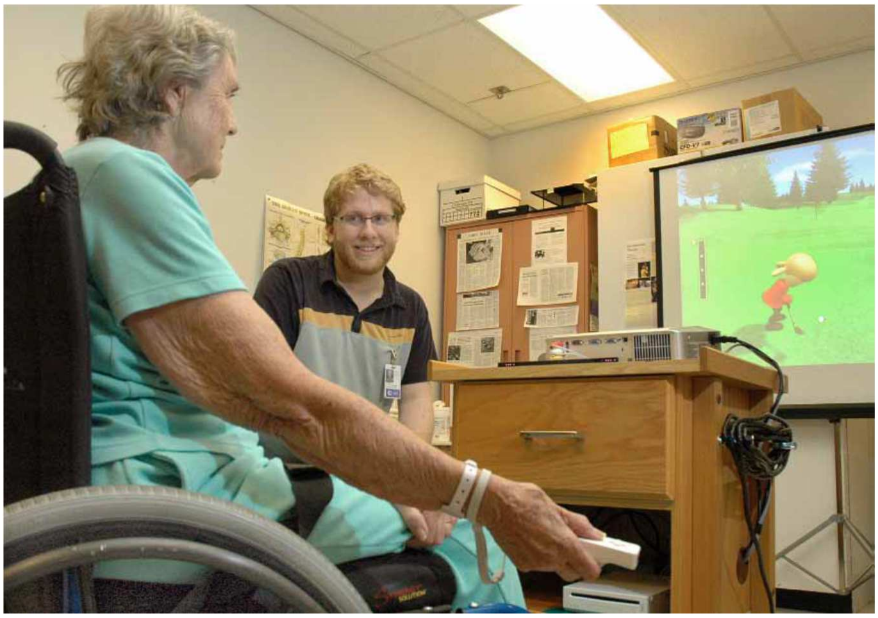


Figure 20: Occupational therapist works with his patient using Wii game console

Other commercial activity-promoting gaming systems than Nintendo Wii (Nintendo) are the Dance Revolution (Konami Digital Entertainment), the Sony EyeToy (Sony Computer Entertainment), and Xbox Kinect (Microsoft), which require player motion and, in some cases, weight bearing to control gameplay. The active nature of these gaming systems may also, offer therapists and patients a number of potential benefits to complement traditional therapies [224].

It has been observed [222] that children with developmental disabilities benefit from Information and Communication Technologies during Occupational Therapy. Some of the advantages of using interactive technologies in OT are listed below [149], [197]:

- **Visual learning style**: there are children with developmental disorders, such as Down syndrome, that have strengths as visual learners and therefore difficulties to learn from listening alone. ICT can provide a source of both visual and auditory stimulations.
- **Non-verbal mode of response**: children with speech and language difficulties often meet difficulties in answering questions verbally. It is easier for them to show their understanding using a non-verbal mode of response, such as a touch of the screen, mouse click or key press.
- **Being in control**: spoken language difficulties can be a source of frustration, which may be expressed through undesirable behavior, because of the feeling being in control. The computer provides an environment where children despite being in

control, and with the appropriate practice can work unsupported, they develop their self-esteem and independence.

- **Opportunities for practice and immediate rewards**: children with disabilities often need more opportunities for practice than their typically developing peers. The computer is able to present infinite chances to try the same activity, reproducing the exact experience over and over again and providing tireless positive feedback in the form of animations, music and sound effects.
- **Errorless learning**: well-designed software can provide activities that are tailored to the individual child's level of skill in a specific area, creating an errorless learning experience. This means that the child is supported by the software, in order to achieve repeated success. Software can be programmed to respond to the child's input and modify the way in which it presents subsequent activities.
- **Self-paced learning**: the child is able to proceed as quickly or as slowly as he or she wishes; the computer will 'wait' for the child to respond without prompting them before they have had time to fully process the information and construct their response.
- **Improving motivation**: a child's attention span may be increased as the learning experience is enhanced with pictures, sounds and animation. One study using interactive commercial software [99], suggested that attention span could be increased from less than 3 minutes to more than 15 minutes in children with mild to moderate learning difficulties.
- **Clutter free working environment**: computer programs can provide a highly organized and predictable working environment that focuses the child on specific learning targets. If needed and according to the child’s needs, software can be flexible to provide an uncluttered and simple layout without a wide variety of distractions and complications.
- **Fear of failure**: children with developmental issues seem to be particularly apprehensive about failure and may demonstrate a range of avoidance behaviors to escape from teaching situations that they perceive as potentially challenging. Therefore, computer software can be carefully programmed to meet individual needs and teaching activities can be graduated in very small stages. Older children soon learn that mistakes can be altered quickly and discretely.

As already mentioned, there are assessment tools that aim to assist therapists in order to monitor developmental disabilities of children. Regarding writing disabilities, an augmented pen has been developed as a tool that supports the therapy of developmental dyslexia, with particular regard to dysgraphia [46]. The augmented pen comprises a display monitor equipped with a high-sensitivity touchpad and specially designed writing tool equipped with pressure sensors (see Figure 21). It measures the pressure put on the surface of the display, eye-hand coordination and whether the pupil holds it properly or not. Additionally, the software of the augmented pen includes a set of activities supporting the treatment of dyslexia, along with a database of pupils and therapy results.

In order to measure the factors mentioned, Smart Pen equips the tablet pen with three pressure sensors. These sensors are mounted on a special grip which helps to encourage the recommended tripod finger position when holding a pen. The grip is similar to the ordinary pen (or pencil) grips used for dyslexic children who often have poor pen grip. Afterwards, the signals from the sensors are translated to the PC format using additional converter with processor built-in A/D converters [46].

The Smart Pen has the main aim to allow a pupil to work under teacher or therapist supervision. According to [46], the application allows continuous monitoring of different parameters related to writing. All these parameters are stored in a database and can be easily reviewed from the therapist in order to observe the therapy progress for each pupil. Furthermore, the Smart Pen software allows preparing new exercise scenarios and in this way closely matching pupils' problems. There are five different activity types possible to choose; drawing, squiggles, coloring, mazes and joining the dots.

The possible parameters that professionals can examine through Smart Pen's software are:

- Pen grip and grips strength measurements
- Pen pressure on the tablet
- time taken to complete an activity
- duration of task completes
- number of times the hand is lifted from the surface, and
- the amount of a user's drawing outside the perimeter.

Monitoring these parameters gives an opportunity to indicate improper performing of the activity.

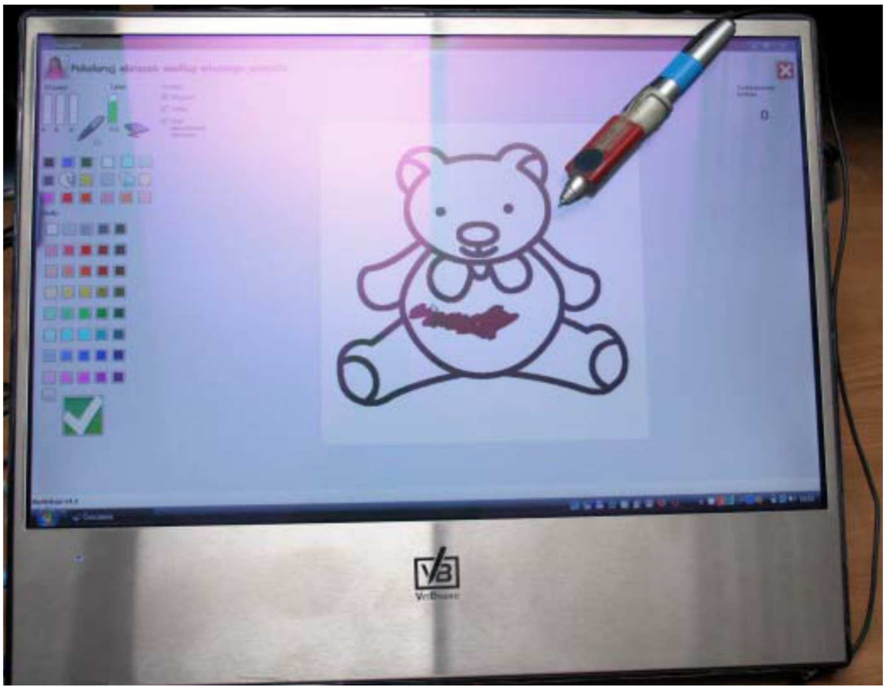


Figure 21: Smart Pen Prototype used in a coloring exercise

2.5.2.2 Activity recognition for monitoring children's developmental progress

In the domain of activity recognition for monitoring children's developmental progress, which is considered extremely relevant to this research work, existing work has invested more on recognition on play. **KidCam** is a prototype system designed to study the use of computer technology to support the early detection of children with special needs [113]. The authors evaluated the ability of KidCam to support parents and pediatricians in the decision-making process to assess if a child was developing typically over a 4-month period. KidCam, a computer supported baby monitor comes with a companion desktop software that allows parents to collect pictures and videos of their child while also providing age-appropriate prompts for parents to enter developmental health-related information about their child.

Figure 22: View of KidCam prototype on a Sony VAIO (left) and a screen shot showing the main menu of KidCam's interface (right)

According to [112], the basic functionality enables the recording of video, audio, and still pictures using either the front or the back camera, as well as reviewing multimedia data based on different annotations that are provided either during or after capture. For instance, parents may witness their children spontaneously take their first steps and wish to go back and record those moments.

Through the supported to KidCam software, parents can watch and monitor the videos as well as the progress captured. They can also conduct information such as the current targets for the selected child based on his/her date of birth, while they can still choose to view different age ranges using the numbered links across the top. The author uses milestones from a standardized list applied in many pediatricians' offices across the United States, called the Ages and Stages Questionnaire [30], which are organized into six categories, including: (a) Communication, (b) Gross Motor, (c) Fine Motor, (d) Personal-Social, (e) Problem Solving, and (f) Overall.

Moreover, **LENA**[39] is a commercial system designed to help monitor language development in children, from new born to four years old. LENA stands for Language

[39] http://www.lenafoundation.org/

Environment Analysis and uses digital signal processing to parse conversation into words. LENA monitors and measures a child's linguistic progress and their language environment by automatically monitoring child vocalizations, words spoken to the child, conversational turn taking, meaningful speech, and exposure to environmental language.

Figure 23: Main screen for viewing a child's milestone progress. Numbered links across the top are used to access different age ranges. The orange box on the left contains milestone questions, and as the parent enters information, it is displayed on the blue panel on the right.

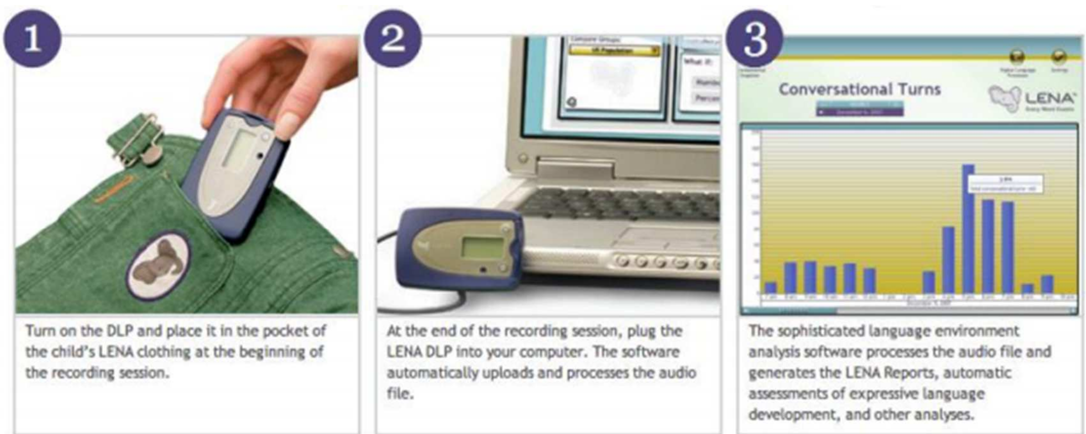


Figure 24: Automatic Language Assessment in Three Steps – LENA

More specific, LENA system includes a small digital recorder that the child wears in a pocket, while recording every sound the child produces for 16-hours. In the meanwhile, it provides a large database processed by language analysis software that uses acoustic properties of the signal that allows researchers to estimate how much talk children produce, how much talk they hear from different sources, and how many conversational turns (i.e., exchanges between two speakers) occur [91]. Researchers using this system have discovered, for instance, that children's language development is related to the number of conversational turns they experience; not just the amount of talk that is addressed to them [250].

The early identification of children with developmental delays is an important public health goal [63]. However, the wide variation of typical development among children can make establishing the presence of a developmental delay, a difficult task. Subtle abnormalities can often be overlooked as normal developmental variation [25]. Therefore, the routine monitoring of a child's progress is crucial to the identification of delays and is considered a vital component of pediatric care [205].

2.6 Progress beyond the state of the art

This section discusses the progress beyond the state of the art of the work presented in this thesis taking into account different perspectives:

- **AmI**: The major contribution form the point of view of AmI is the production of a complete framework that optimally addresses the requirements of children playful interaction within the intelligent environment. The AmI environment is enhanced with novel interaction metaphors and techniques in the context of AmI edutainment scenarios. To achieve this, this thesis presents the design and implementation of a technological framework to support the interaction requirements of AmI at large, by offering a number of alternative natural interaction techniques such as gestures, face and skeleton tracking, head position estimation and speech recognition.
- **Ubiquitous Computing**: Ubiquitous Computing is employed for the integration of ICT into everyday objects and the facilitation of their integration and interaction with the environment through a service oriented architecture facilitated by a pervasive middleware. Basic contribution in this context is the development of a set of service specifications to achieve low level integration of heterogeneous devices, sensors, applications and services in AmI environments.
- **Augmented and Mixed Reality**: The research work presented in this thesis integrates several physical objects either as augmented physical artifacts or computer vision recognized physical artifacts. These artifacts together with a number of games contribute to the creation of novel mixed reality applications especially designed for children.
- **Intelligent User Interfaces**: The developed framework is pioneer regarding the usage of environment information to achieve interaction monitoring. Furthermore, such runtime information is used for the first time not only for optimally adapting the interaction metaphors to the requirements of each user, but also to extract valuable information regarding whether the child meets the typical of his age developmental milestones.

- **Knowledge representation and extraction**: Regarding knowledge representation and extraction, the innovative aspect relies on the fact that the framework integrates an enriched dynamic profile model of children's functions and skills based on the ICF-CY classification. Based on such model, and using interaction monitoring data and appropriate statistical analysis techniques, the proposed framework is able to produce an accurate representation of child's current developmental status, as well as recommendations regarding further investigation by child developmental experts. To this end, parents are provided with general information (in a pleasant and aesthetic way) about their child's physical and mental development progress, as well as indications of a possible skill immaturity. Furthermore, early intervention professionals are provided with extensive data in addition to the full interaction history for reasoning about whether the child is meeting all the necessary developmental milestones. At the same time, this technological infrastructure is employed with the help of occupational therapists to create innovative ways to use new technologies in their practice.
- **AmI games**: The proposed framework facilitates the creation of smart games that share unique characteristics such as: a) being adaptive to children's skills and abilities, b) being able to establish the appropriate communication channels with children (verbal or non-verbal) c) allowing children to have the control so that with practice being able to play unsupported and thus developing their self-esteem and independence, d) improving motivation, and e) reducing fear of failure. The proposed infrastructure facilitates the automated extraction of knowledge based on interaction monitoring to offer indications regarding the children developmental state, maturity level and skills.
- **Three-dimensional virtual partner**: The proposed infrastructure includes a novel, remotely controlled three-dimensional full body avatar framework for the purposes of edutainment and instructor-student interaction. The avatar allows the implementation of novel multi-presence gamified educational scenarios in multiple desktop computers and mobile devices. As far as anyone, such an avatar framework does not exist yet in an open, research-oriented environment and could be achieved only with closed game-engines and commercial off-the-shelf (COTS) components that do not allow diversity, and extensibility.
- **ICT in OT practice**: The provided framework allows the detection of potential children development delays to be further investigated and diagnosed if necessary. To this end, parents are provided with general information (in a pleasant and aesthetic way) about their child's physical and mental development progress as well as indications of a possible skill immaturity. Furthermore, early intervention professionals are provided with extensive data in addition to the full interaction history for reasoning about whether the child is meeting all the necessary developmental milestones. At the same time, this technological infrastructure is employed with the help of occupational therapists to create innovative ways to use new technology in their practice.
- **FSM description language:** In the context of this research work a new scripting language, called ACTA, was developed so as to facilitate the activity analysis process during smart game design by early intervention professionals who are not familiar

with traditional programming languages. Furthermore, developers can use ACTA not only for developing event-driven sequential logic games, but also for applications of behavior composed of a finite number of states, transitions between those states, and actions as well as for application based on rules driven workflows.

3 Design of a framework to support young children development in AmI environments

3.1 User groups and requirements

This section aims at specifying the target user groups that were addressed by this research work. These groups include young children, who are the main recipients, as well as people surrounding them and taking care of them, such their parents (or other relatives) and early intervention professionals (i.e., occupational therapists). Furthermore, this work aims to facilitate the process of design and creation of smart applications and games for young children. To this end, developers are considered to be another target user group of the current work. This user base sets specific requirements for the applications to be developed, especially in relation to safety, usability, learnability, etc. Each of these requirements encompasses a specific user goal that should be achieved through the usage of these applications.

3.1.1 User groups

This section identifies the stakeholders addressed by the present research work, analyzes the basic fields of interest and makes an analysis of each user group involved.

3.1.1.1 UG1 - Young Children

This user group consists of preschool children aged from 3 to 6 years old. During this period, their main interest is play. Undoubtedly, play is not a waste of time for children. On the contrary, it contributes to overall development [79].

Firstly, play enhances child's physical and motor development, which involves height, weight, general appearance of the body, and coordination of large and finer muscles. During toddler hood, the child learns to climb ladders, balance, dig, pull, and push objects. Threading beads and holding toys are other activities that promote physical and large motor development. Drawing, painting, scribbling and writing enhance eye-hand coordination and finer muscular development.

Secondly, the child's cognitive development includes forming self-concept and forming concepts of size, shape and colors. By playing with different types of blocks, beads, toys, clay, sand, puzzles and other natural materials, children learn to differentiate between different shapes, size, color, texture and volume, which are prerequisites of reading and learning numbers [88].

Thirdly, social and emotional development includes establishing relationships, developing behavioral controls and social skills that make children accepted members of their family, school and community. Play activities such as singing, creative story-telling, drama, role play and puppetry help children in relating with others and in recognizing their own feelings and those of others. Finally, the activities described above, including word games, riddles and group games encourage language development of children.

For the purposes of this thesis, it is important to identify not only the current developmental level of a child, but also to support the child to progress to more advanced skills, concepts and abilities. In order to support young children skills and abilities, appropriate amounts of support should be given at proper times during playing. This approach is similar to the concept of *scaffolding* under which aiming instruction within a child's *Zone of Proximal Development* (ZPD) can promote the child's capacities and development [230].

3.1.1.2 UG2 - Parents

The user group (UG2) includes 'parents' and in more general persons who are responsible for raising a child. Parents make sure that children are healthy and safe, equip them with the skills and resources to succeed as adults, and transmit basic cultural values to them. Parents offer their children love, acceptance, appreciation, encouragement, and guidance. They provide the most intimate context for the nurturing and protection of children as children develop their personalities and identities and also as they mature physically, cognitively, emotionally, and socially. Parents pursue a harmonious development for their child. Undoubtedly, most parents worry about whether their child is developing normally or not, i.e., is meeting all the necessary developmental milestones according to his age.

It is well documented that the proper role of the parent in child development is to provide encouragement, support and access to activities that enable the child to master key developmental tasks. A parent is their child's first instructor and is vital for the wellbeing of a child to be exposed to age appropriate challenges, as well as to experiences that allows the child to explore on their own and learn from interacting with their environment. The authors of [92] claim that the way a child turns out can be determined in large part by the day-to-day decisions made by the parents who guide that child's growth. A developmental trajectory describes the course of a behavior over age or time. Effective parents observe, recognize and assess their child's individual genetic characteristics, then cultivate their child's strengths. According to [92], parents can guide their children's development in four complex and dynamic ways:

- Parents initiate trajectories, sometimes trying to steer their child towards a preferred developmental path based, on either the parents' preferences or their observations of the child's characteristics and abilities, such as enrolling their child in a class, exposing them to people and places, or taking a child to practices or lessons.
- Parents also sustain their child's progress along trajectories with encouragement and praise, by providing material assistance such as books, equipment or tutoring, and by allocating time to practice or participate in certain activities.
- Parents mediate trajectories, which influence how their child perceives and understands a trajectory, and help their child avoid negative trajectories by preparing the child to deal with potential problems.
- Parents react to child-initiated trajectories.

There are other ways parents influence a child's progress on a trajectory, such as through modeling desired behaviors, or modifying the speed of development by controlling the type and number of experiences.

3.1.1.3 UG3 – Early Intervention Professionals

Decades of rigorous research show that children's earliest experiences play a critical role in brain development [80]:

- Neural circuits, which create the foundation for learning, behavior and health, are most flexible or "plastic" during the first three years of life. Over time, they become increasingly difficult to change
- Persistent "toxic" stress, such as extreme poverty, abuse and neglect, or severe maternal depression can damage the developing brain, leading to lifelong problems in learning, behavior, and physical and mental health
- The brain is strengthened by positive early experiences, especially stable relationships with caring and responsive adults, safe and supportive environments, and appropriate nutrition
- Early social/emotional development and physical health provide the foundation upon which cognitive and language skills develop
- High quality early intervention services can change a child's developmental trajectory and improve outcomes for children, families, and communities.

Early childhood intervention is important because it allows children with delays or disabilities to become full and active members of their societies. There are various situations where a child should be referred for early intervention services. These include the following circumstances:

- A child is not meeting expected developmental milestones
- Parents are concerned about their child's development or behavior
- Children are diagnosed with a specific disability at birth that will likely result in delays
- Children experience malnutrition, trauma, very low birth weight, and certain medical or surgical procedures
- Physicians, teachers, or other professionals are concerned about a child's development.

Intervention services help children develop different skills including physical, communication, and cognitive skills, and provide support for self-help, as well as social, emotional and behavioral development. They help children achieve better outcomes, support families, and save precious resources for the community.

Professionals who are involved in early intervention[40] include:

- Occupational Therapists, referred to as OT, are involved in the "occupation" of childhood - daily living skills and play.

[40] http://www.heartland.edu/heip/professionalsAndProvidersInEarlyIntervention.jsp

- Physical Therapists, many times referred to as PT, are involved in the large muscle movements such as crawling, walking, and throwing a ball, as well as more refined motor movements such as grabbing toys, buttoning clothing, and using a crayon.
- Speech Language Pathologists (SLP) are involved with the acquisition of speech and language, feeding issues, and oral motor control.
- Audiologists are concerned with a child's ability to hear and process the auditory world. Psychologists are involved in assessing a child's development through a variety of psychological tests and assisting families in understanding their child's development.
- Social Workers help the family to understand the emotional and social aspects of their child and may provide counseling and support to help families cope with a variety of stressors.
- Registered Dietitians work with a family on the nutritional needs of the child.
- Nurses provide information and support surrounding health and wellness issues.
- Family Therapists provide services to help families understand and accept the disability and encourage them to fully develop their child's potential.
- Child Development Specialists provide expertise in the area of child development, helping families to understand the sequential nature of development and how best to promote their child's development.
- Vision Specialists deal with all areas of a child's vision.

According to [35], early intervention occupational therapy practitioners use tele-health technologies to provide services for children participating in early intervention services. Benefits include increased access to occupational therapy services for children who live in remote areas, the prevention of unnecessary delays in receiving services, and coordinated care among team members in different locations.

3.1.1.4 UG4 – Game Developers

Games developers are involved in the creation and production of games for personal computers, games consoles, social/online games, arcade games, tablets, mobile phones and other hand held devices. Their work involves either design (including art and animation) or programming. In detail, there are many stages including creating and designing a game's look and how it plays, animating characters and objects, creating audio, programming, localization, testing and producing. Typical activities of game developers may include:

- developing designs and/or initial concept designs for games including game play
- generating game scripts and storyboards
- creating the visual aspects of the game at the concept stage
- using 2D or 3D modelling and animation software, such as Maya, at the production stage
- producing the audio features of the game, such as the character voices, music and sound effects
- programming the game using programming languages such as C++

- quality testing games in a systematic and thorough way to find problems or bugs and recording precisely where the problem was discovered
- solving complex technical problems that occur within the game's production;
- disseminating knowledge to colleagues, clients, publishers and gamers
- understanding complex written information, ideas and instructions
- working closely with team members to meet the needs of a project
- planning resources and managing both the team and the process
- performing effectively under pressure and meeting deadlines to ensure the game is completed on time.

In this thesis, the user group 'game developers' (UG4) consists of those developers who have the following responsibilities: a) generating game scripts, b) programming the game using programming languages, and c) working closely with team members to meet the needs of a project.

3.1.2 User requirements

User requirements analysis provides precise descriptions of the content, functionality and quality demanded by prospective users. There is a wide variety of methods and techniques for user requirements elicitation, mostly originating from the social sciences, psychology, organizational theory, creativity and arts, as well as from practical experience [129]. Many of these techniques are based on the direct participation of users or user representatives in the process of formulating their own technological needs.

Methods such as observation, interview, document analysis, focus group and analysis, checklists or questionnaires can be used for the elicitation of user requirements. The process of requirements identification and analysis, sets a variety of techniques under discussion in order to find out which are most suitable for the afore-mentioned user groups UG1, UG2, UG3, and UG4. These techniques are listed together with their benefits and drawbacks in the following table [10].

✓ *Appropriate*
■ *Needs modifications and adjustments*
☒ *Not recommended*

		Disability				Age	
		Motion	Vision	Hearing	Cognitive/ Communication	Children	Elderly
User requirements Elicitation Methods and techniques	Brainstorming	✓	✓	■	■	■	■
	Direct Observation	✓	✓	✓	✓	✓	✓
	Scenarios, Storyboards and Personas	✓	✓	✓	✓	✓	✓
	Activity Diaries and Cultural Probes	■	■	✓	■	■	✓
	Surveys and Questionnaires	■	■	■	☒	■	■
	Focus Groups	✓	✓	■	☒	■	■
	Interviews	✓	✓	■	☒	■	■
	Empathic modeling	✓	✓	✓	☒	☒	☒
	Prototyping	✓	✓	✓	✓	✓	✓
	Cooperative, Participatory Design	✓	✓	✓	■	■	■

Table 7: Summary of user requirements elicitation methods and techniques

Therefore, the most suitable methods and techniques adopted for user requirements elicitation (taking mostly into account the specific user group UG1) are the following: a) Direct observation, b) Scenarios, Use Cases and Personas, and c) Prototyping. In the case of UG1, user requirements were mainly gathered through the elicitation process of observing young children playing in a kindergarten, at home, etc., or during occupational therapy treatment. Additionally, during playing, children were given early prototypes of augmented artifacts such as toys, pen, etc., as well as touch screen mobile games for kids found on the market. The observation data were used to envision various usage scenarios presented in section 3.1.3. Brainstorming elicitation method was used to gather the user requirements of the remaining user groups UG2, UG3, and UG4.

For the identification of user needs, the user perspective was assumed, resulting in functional requirements and non-functional requirements. Functional requirements consist of the goals that users want to reach and the tasks they intend to perform, while non-functional requirements are constraints on various attributes of these functions or tasks associated with the user's goals.

The following sections present the functional and non-functional requirements for the aforementioned stakeholders towards the creation of a framework (**smart games**, **authoring tools**, and **monitoring applications**) to support young children development in AmI environments.

3.1.2.1 Functional requirements

Functional requirements capture the functionality required by the system towards effectively supporting (an agreed upon set of) user tasks. In other words, functional requirements define what a system is supposed to do, i.e., "**system shall <do requirement>**". For the purposes of this study, the word "**system**" represents the entire proposed framework to support young children development in AmI environments.

3.1.2.1.1 UG1 – Young Children

Play activities are essential for children's healthy development. These activities influence the development of fine and gross motor skills, language, socialization, personal awareness, emotional well-being, creativity, problem solving and learning ability.

- **UG1-R1**: The system should enhance through play children's experience with a wide variety of content (art, music, language, science, math, social relations) because each content type is important for the development of a complex and integrated brain.
- **UG1-R2**: The system should support play that links sensory-motor, cognitive, and social-emotional experiences, which are all essential for an ideal setting for brain development.
- **UG1-R3**: Most children have little trouble expressing their creativity. The system should support toys that allow a child to get creative, encourage imagination and allow him to learn by doing.
- **UG1-R4**: The system should enhance abilities such as imagination, thinking, and problem solving.

- **UG1-R5**: The system should provide smart games that:
    i. can be used in many ways
    ii. allow children to determine the play
    iii. appeal to children of more than one age group or level of development
    iv. can be used together with other games to produce new ways of playing
    v. by timeless and continue to be part of play as they develop new interests and skills
    vi. help children develop skills important for further learning and a sense of mastery.
- **UG1-R6**: The system should support the child's self-esteem. To this end, it is important for children to set goals, develop a plan to meet that goals, and complete tasks while playing. Small, uncomplicated tasks are the best strategic way to begin with. Praise must to be given throughout the task and not only upon completion of every task. Additionally, the system should offer games which properly support children to overcome potential failures without feeling punished.
- **UG1-R7**: The system should provide children with amusing and entertaining games as well as communicate with them in a natural and attractive way. Additionally, the system should provide games consisting of attractive physical toys and offering appealing aesthetics for their children.
- **UG1-R8**: The system should make easy for the child to interact by himself, (i.e., without the presence of an adult during play). For example, the system should offer games which provide children enough information in order to start playing, and also sensitive help to continue. Additionally, the system should interact with children without any safety concerns during play.

3.1.2.1.2 UG2 – Parents

Parents have to play various roles in their child's life. One of the most important roles is to provide them an environment full of love and incentives. Another role is to help the child develop a positive self-image. Most parents worry about whether their child is developing normally or not, i.e., whether they are meeting all the necessary age-related developmental milestones. As a result, the system should satisfy the following requirements:

- **UG2-R1**: The system should provide parents with information related to their child's physical and mental development progress.
- **UG2-R2**: The system should notify parents about any possible delayed ability.
- **UG2-R3**: In case of delayed ability, the system should suggest parents about game tasks to focus their attention during their child's playing.
- **UG2-R4**: The system should report parents usage statistics related to their child playing frequency of selected games, as well as the rate of successful (or unsuccessful) game completion.

3.1.2.1.3 UG3 – Early Intervention Professionals

Child's profile plays a very important role for early intervention professionals when children are referred to them for assessment or treatment. They commonly obtain information about a child's health, progress, development and behavior, as well as his strengths and weaknesses in a number of areas, through the use of specific questionnaires.

- **UG3-R1**: The system should keep information about child's development progress with extensive data as well as full interaction history.
- **UG3-R2**: The system should monitor and assess a large set of child's skills and abilities found in the ICF-CY that can be used to extract indications about child's development progress.
- **UG3-R3**: The system should provide early intervention professionals with instant access to the child's related information.
- **UG3-R4**: The system should notify early intervention professionals for indications of child's delayed ability.
- **UG3-R5**: In case of delayed ability, the system should suggest early intervention professionals about game tasks to focus their attention during child's playing
- **UG3-R6**: The system should report early intervention professionals usage statistics related to their child playing frequency of selected games, as well as the rate of successful (or unsuccessful) game completion.
- **UG3-R7:** The system should provide early intervention professionals with details about a delayed ability, as well as the involved body structures accompanied by ICF-CY codes for clinical practice.

3.1.2.1.4 UG4 – Game Developers

Under the perspective of game creation in AmI environments, the following set of requirements was identified as necessary for game developers:

- **UG4-R1**: The system should maximize the potential for incorporating heterogeneous technologies
- **UG4-R2**: The system should provide sensible programming abstractions
- **UG4-R3**: The system should enable the implementation of higher-level functionalities by composing services
- **UG4-R4**: The system should provide a flexible model for representing and using the context of the AmI environment supporting young children's development
- **UG4-R5**: The system should provide libraries and tools for developing and deploying service.
- **UG4-R6**: The system should provide libraries and tools for developing and deploying games suitable for young children
- **UG4-R7**: The system should provide the means for simulating the functionality of the AmI environment supporting young children's development
- **UG4-R8**: The system should provide libraries and tools for developing and deploying application to monitor young children's development.

3.1.2.2 Non-functional requirements

Non-functional requirements are constraints on various attributes of the system's functions or tasks associated with the users' goals. In other words, non-functional requirements specify the required behavior of the system towards supporting the user goals efficiently, i.e., the user might end up disliking the system if some of his/her goals (e.g., to perform administrative tasks as quickly as possible; to stay away from big mistakes, to be consistent) are violated while using the system. The most relevant non-functional requirements supporting the aforementioned stakeholders are the following:

- **NF-R1**: **Learnability**. It is essential for each application incorporated in the Ambient Environments to be developed to require a minimum level of learning from the user. To this end, the provision of easy to use, self-descriptive and intuitive user interfaces constitute a fundamental requirement.
- **NF-R2**: **Satisfaction**: Offering satisfaction to the user by accessing an interactive application is considered a primary requirement. More specifically, this is achieved by offering support for recovering from user errors and furthermore via offering an overall feeling of smooth operation and facilitating the user to perform complex operation through an intuitive and usable interface.
- **NF-R3**: **Robustness**: The application must be extremely robust against all kinds of mis-use and errors. Wrong inputs must not lead to a system malfunction or crash.
- **NF-R4**: **Availability**: The application must do its job even in the presence of hardware component crashes, shortage of hardware resources such as storage or communication bandwidth, and other exceptional conditions.
- **NF-R5**: **Extensibility**: The application incorporated in the AmI environment must support its extension by new hardware or software components at runtime. However, this requires further investigation from the field of standardization for offering common hardware and software standards for allowing the cooperation of applications developed for specific Ambient Infrastructures.
- **NF-R6**: **Safety**: Safety requirements are the ‘shall not’ requirements which exclude unsafe situations from the possible solution space of the system.
- **NF-R7**: **Security**: An AmI environment system, although continuously monitoring persons, must guarantee a well-defined degree of privacy for the persons under observation. The privacy rules must be precisely formulated and verified.
- **NF-R8**: **Timeliness**: Most services in an ambient environment system, have to be carried out in real time, such as the emergency treatment. Long propagation delays after the detection of an emergency are not tolerable.
- **NF-R9**: **Resource Efficiency**: The available resources, i.e., processing power, memory, communication bandwidth and energy, have to be utilized as efficiently as possible in order to allow: a) an affordable price of the systems, and b) the realization of highly integrated, autonomous sensor nodes with a high endurance, which is of particular interest if the sensor nodes have to be mobile.
- **NF-R10**: **Natural, Anticipatory Human-Computer Interaction**: Ambient Intelligence environments have to provide human interfaces for current user groups. Each group has different requirements for interacting with the system. Multimodal interaction paradigms that combine several modes are a powerful approach to enhance usability. Anticipatory interfaces, which proactively contact persons in certain situations, are considered mandatory.

- **NF-R11**: **Adaptivity**: The systems are able to adapt themselves at runtime. Adaptivity on different levels and scales is considered an outstanding characteristic of living assistance systems.
- **NF-R12: Fun:** This denotes the need to make something fun, engaging, or enjoyable. Systems are no longer being designed solely as functional or productivity devices. Most commonly, they seem to be a combination of entertainment and function (i.e. mobile phones, edutainment).

Based on the identified situations, the system can perform:

- **NF-R2: Self-optimization**: This denotes the ability of the system to adapt its algorithmic behavior to changing needs of the application. An example of self-optimization is the dynamic increase of the volume of loudspeakers in the case of existing environmental noise.
- **NF-R2: Self-configuration**: This denotes the ability of the system to dynamically integrate new software components and remove existing ones not needed any more. Self-configuration is a form of self-adaptation at the architectural level of a system. Self-reconfiguration may be triggered by changes in the hardware configuration aiming at a better use of resources or a higher degree of fault tolerance.
- **NF-R2: Self-maintenance**: This denotes the ability of the system to perform standard maintenance tasks such as downloading new updates and releases automatically from a remote service center.

3.1.3 Scenarios of use

In this section, narrative descriptions of interactive processes will be presented, including user and system actions and dialogues. More specifically, a set of realistic examples will be given regarding what tasks the users can carry out in the specified context of the system.

3.1.3.1.1 Young children's edutainment

Dimitris is a three years old child and Max is an interactive virtual partner. The role of Max is to communicate with Dimitris in various modes such as didactic or playful. Dimitris, with the presence of Max, plays while developing his learning skills through entertainment with games such as building blocks or puzzles. Interactive games with object manipulation affect Dimitris' cognitive development positively, by enhancing his creative thinking.

Dimitris enjoys playing with jigsaw puzzles, one of his favorite games. Max suggests Dimitris to play a few puzzles more appropriate for his age. Max records the interaction history, i.e., the way that Dimitris tries to match the adjacent pieces based on their shape or their displayed picture. Max often rewards Dimitris, so that he gains confidence in himself and in his abilities.

Puzzles are one of the most appropriate game categories for enhancing cognition, hand-eye coordination, reasoning and knowledge. However, Dimitris prefers to play, with

Max's presence, interactive games from different categories (colors, numbers, animals etc.).

3.1.3.1.2 A young child playing with matching cards

John is a young child who loves playing with matching cards. When his mother gives him a matching card game he immediately starts observing the cards and after a while, he is capable of selecting successfully the matching pairs. For example, he places the card presenting an engineer next to another card showing the engineer's various tools. Afterwards, he takes both cards and places them aside, indicating that he has finished with them. Unfortunately, John did not notice that the card displaying the tools was rotated by 180o degrees and continued trying to match the remaining cards. The matching process is being completed when John matches all the cards despite the fact that some of them may be upside down. However, if this mismatch problem happens several times, it may possibly indicate the existence of an underlying weak ability for "orientation in space". The proposed system should be able to monitor children's play and extract indications of these delayed abilities.

3.1.3.1.3 A young child with hyperactivity

Peter is a seven years old boy who often fidgets with his hands or feet or squirms when seated and has difficulty in playing. He likes playing but he has difficulty keeping still of time and he doesn't seem to be able to think before acting. When he starts playing a game, with his mother in the role of a playmate, he acts very quickly usually without waiting to hear the instructions, while simultaneously clapping his hands in excitement. Peter fails to wait for his turn in games and sports and he rides his bicycle into a busy street without looking. His parents claim that Peter is just a very energetic child. However, Peter has some symptoms of hyperactivity and inattention. Therefore, through the monitored play, the proposed system should be able to extract the appropriate indications and suggest that the child's parents should refer him to a clinic for further evaluation.

3.1.3.1.4 A young child with lack of communication

Dimosthenis is a three years old boy who loves playing games with farm animals, while hearing their voices. As Dimosthenis starts playing an interactive farm game, the system firstly requires Dimosthenis, through a virtual character, to put his identification card on the surface in order for the game to begin. The boy is very quiet and does not seem to react at the request of the system. After many attempts, the systems recognizes that there is an understanding issue that obstructs the boy to interact with the game and start playing.

Nevertheless, in order for the system to assist Dimosthenis comprehend what exactly he expects from him, it illustrates a video with the movements that the boy should follow in order to make the game begin. Ultimately, the boy interacts with the game and starts playing happily by placing the animals into the farm with no assistance from adults. Dimosthenis appears to have some problem of communication, something that the proposed system is able to monitor, so it thus suggests the need of treatment from early intervention professionals.

3.1.3.1.5 Remote monitoring with indication of imitation dysfunctions

The parents of Ioanna have the feeling that their child has unspecified disorders and thus they consult an occupational therapist. The therapist suggests letting Ioanna, a five years old girl, to play with the interactive table the mimesis game. This game, with the help of a virtual character, allows children to make specific movements according to instructions. Through this game, the occupational therapist is able to monitor and indicate skills such as: a) motor planning, b) the ability to imitate postures, c) reception of language, d) visual perception, and other.

In order for the therapist to draw a conclusion, Ioanna must play more than once in ideal circumstances. For that reason the therapist suggests Ioanna's parents to get her an interactive table and place it at home, in order for the child to feel free with no limitations or stress. In this way, Ioanna can interact with the virtual character at home and try to imitate his postures. On the other side, the occupational therapist is able to monitor her performance remotely at various times and extract through the system a report with indications of disorders as emerging through Ioanna's interaction with the table.

3.2 Ambient design

The goal of Ambient Intelligence is to be mixed in the fabric of human activities acting as a non-visible computerized framework. From this point of view, Ambient Intelligence must and should be incorporated into human space non-intrusively and gracefully. Therefore, it is essential to investigate not only the technological aspects, but also the analysis of the structure and architecture of the physical space where human activities take place and are facilitated by the technological environment.

This section presents the environment setup (in terms of structure and architecture) aiming to facilitate the usage of the proposed technologies in the context of real life scenarios. Specific attention is paid to: a) the designing of the AmI simulation space including the augmentation of everyday objects, b) the methodology for creating smart games that monitor young children's abilities and automatically adapt to them, and c) the creation of the necessary toolkits to assist the easy integration of the proposed technologies in the process of game creation, and d) the creation of the necessary administration facilities to help parents and early intervention professionals to monitor young children's development progress.

3.2.1 Design of the AmI simulation space

In the context of AmI, interactive devices must be unobtrusive, hidden or embedded in traditional everyday objects and furniture augmented with ICT technology without compromising general health and safety requirements. To support the goals of the current study, an artifact-oriented approach has been adopted. The latter introduces independent AmI augmented artifacts in an environment surrounding young children (e.g., a child room or a kindergarten) taking into account that it is important for the equipment to be easy to install, easy to move around in the room and shouldn't take up too much space. The 3D representation of a proposed setup of a room equipped with augmented artifacts is presented in the following figure (see Figure 25).

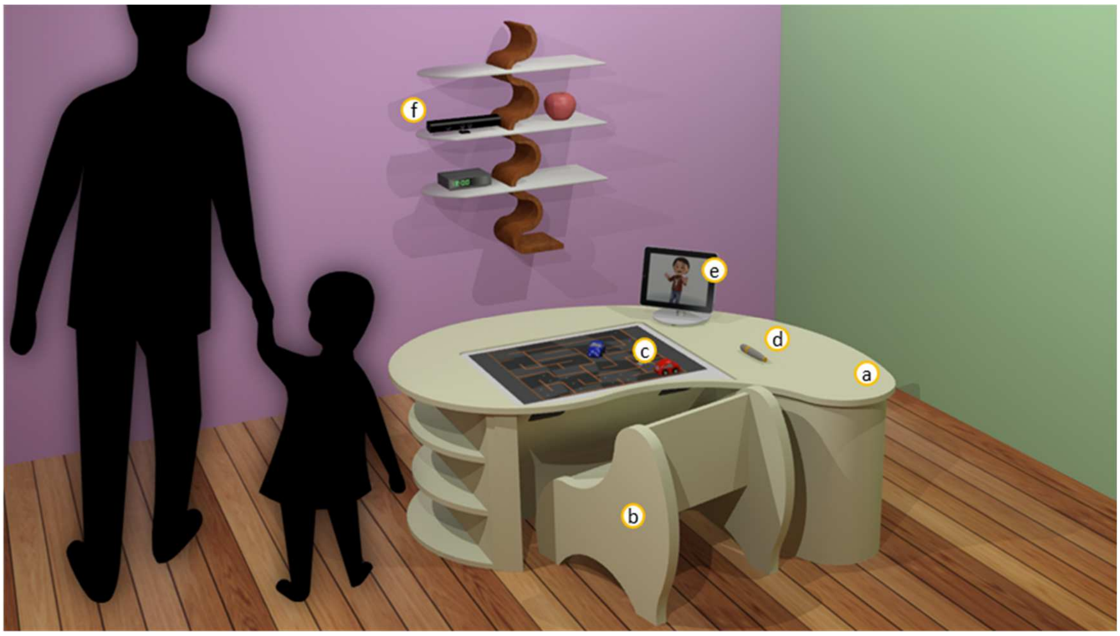


Figure 25: A 3D representation of a room equipped with augmented artifacts

The augmented artifacts depicted in Figure 25 are: a) an augmented interactive table suitable for young children, b) an augmented chair, c) a main display surface with tangible objects on it, d) an augmented pen, e) a secondary display device and, f) a casual wall shelf unit equipped with a special sensor module. The following sections describe such artifacts in details.

3.2.1.1 Augmented interactive table

The first artifact depicted in Figure 25 (1) is an augmented interactive table suitable for young children aged between 3 and 6 years old. According to the physical measures of a child's size, form, and functional capacities, the table's height does not exceed 45-46 cm [57]. The primary goal of the design is the augmented table to be robust and safe avoiding sharp corners. On the one hand, that design allows young children to play freely on the table's surface, while, and on the other hand, the table could be used as any standard table for eating, drinking, and other common activities.

A main display device is embedded in an invisible way on the top of the augmented table. The main display device (see Figure 25 (3) and Figure 26) is enabled with multi-touch and force-pressure sensitive capabilities, and simultaneously is able to recognize the location and the rotation of physical objects. Physical object recognition is be achieved through the use the embedded cameras (i.e., to get clear images of objects that are left on the surface) and the appropriate vision software components. To this end, the main display panel is made from a high robust translucent surface covered with window frosted film. Therefore, using back projection, the translucent surface can be used for the projection of the user interface and, at the same time, the recognition of special visual markers placed on the abutting side of each physical object.

Even though back projection introduces space-related requirements, it eliminates the need for the installation of ceiling mounted cameras or hanging projectors. In case of mounted cameras or hanging projectors, special visual markers had to be placed on the top side of each physical object resulting into an unnatural multi-touch interaction experience. The resolution of the main display device of dimensions of 56cm (L) X 40cm (W), depends on the native resolution of the embedded projector.

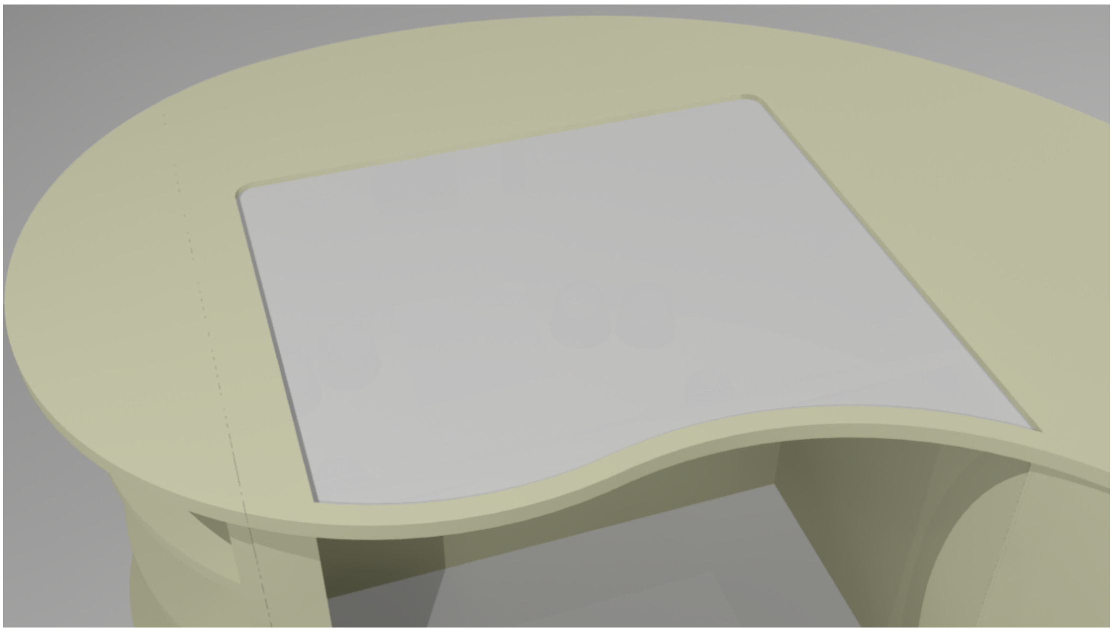


Figure 26: The main display device

The hardware components required for the operation of the aforementioned applications (i.e., multi-touch, object recognition, force pressure recognition, etc.) are embedded and non-visible to the user. Figure 27 shows all the necessary hardware components installed inside the artifact. In detail, the embedded hardware infrastructure includes:

- A high performance computer equipped with high end graphics card
- A mini sized portable led projector with low heat emission and low operating noise level
- A mirror for reducing the projection distance
- Two high quality cameras located behind the main display device equipped with wide lens to maximize the quality of the captured image
- Two or more infrared beacons located behind the main display device covered with thin fabric or mat film to diffuse the light smoothly
- Two or more cooling fans located at specific points that facilitate the natural air flow.
- Two or more temperature sensors
- Stereo speaker set
- Pressure sensors located just behind the screen.

In order to determine the exact dimensions of the table a series of tests were conducted. The augmented table had to be as convenient as possible and safe avoiding sharp corners, while it should also provide enough space to install the aforementioned necessary hardware components, as well as allow sufficient space to accommodate the child's legs (see Figure 27 and Figure 30).

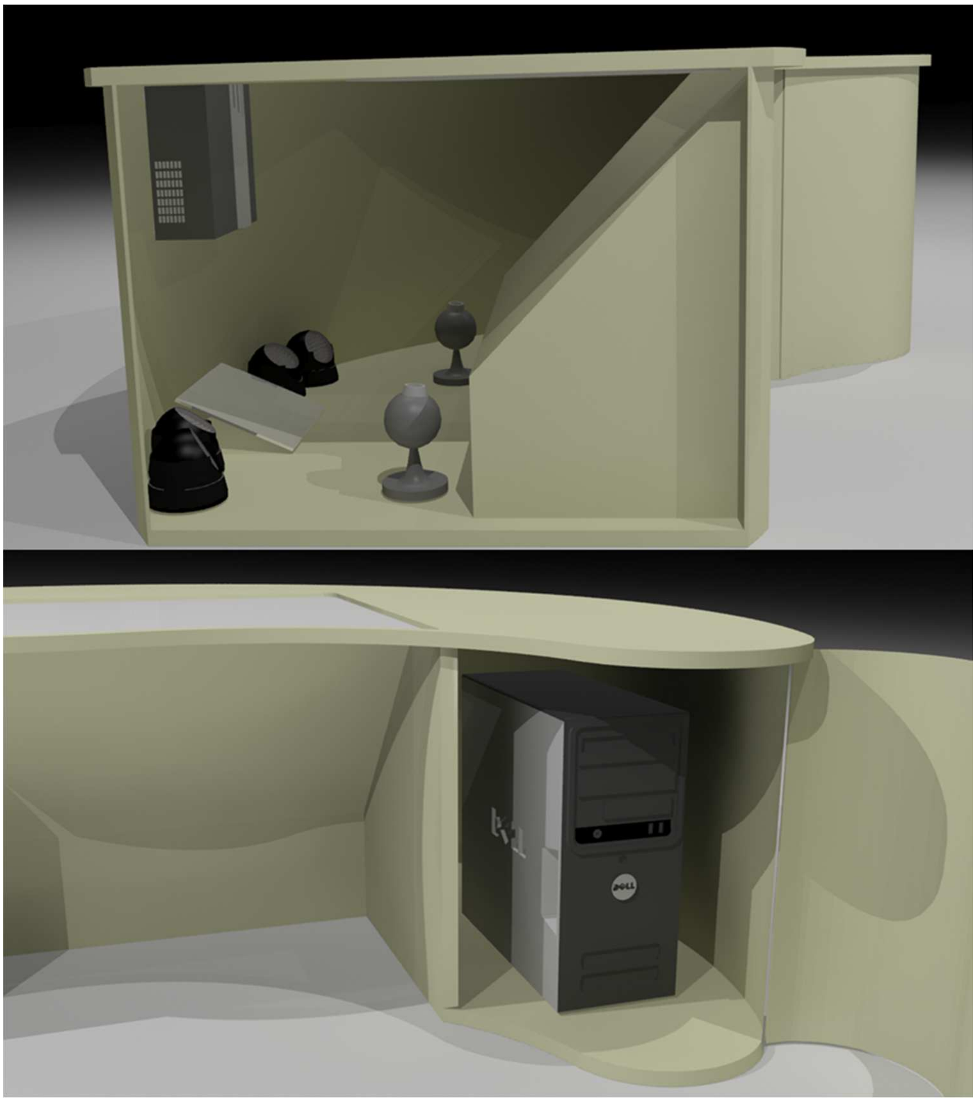


Figure 27: The hardware infrastructure

Regarding the hardware selection, the choice of the appropriate projector was done based on various criteria such as: a) the distance to the projection pane which had to be minimized, b) the type of the projector's lamp (i.e., requiring as little maintenance as possible), and c) the size of the projector itself which should be as small as possible. Because of the table's height (which should not exceed 45-46 cm), the main selection criterion of the cameras was the angle view and therefore cameras with wide lens were preferred.

Having chosen the LG HX300g LED Projector, a rough prototype was implemented in order to measure the exact table dimensions and confirm that the visual software components captured the entire surface and recognized successfully the physical objects placed on it (see Figure 28). After numerous iterations, the exact dimensions of the interactive table were determined. As depicted in Figure 29, the table's dimensions were measured to be 116cm (L) X 105cm (W) X 46cm (H).

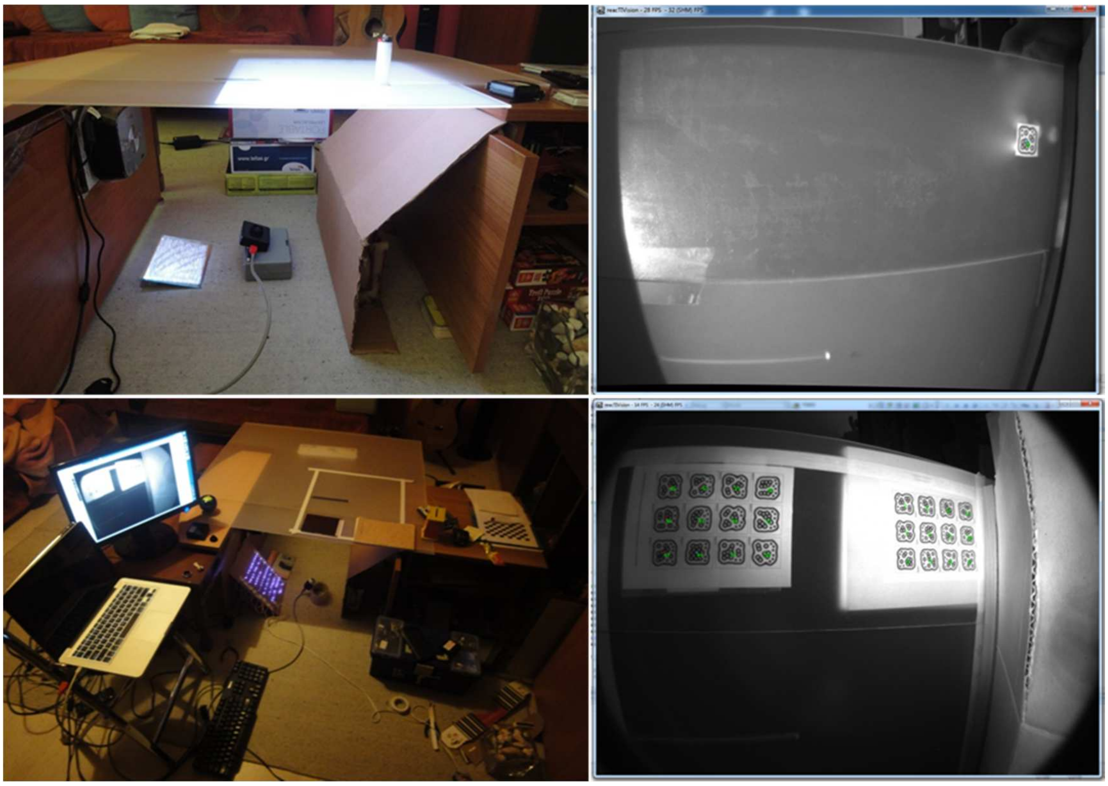


Figure 28: Various prototype setups to determine artifact's dimensions

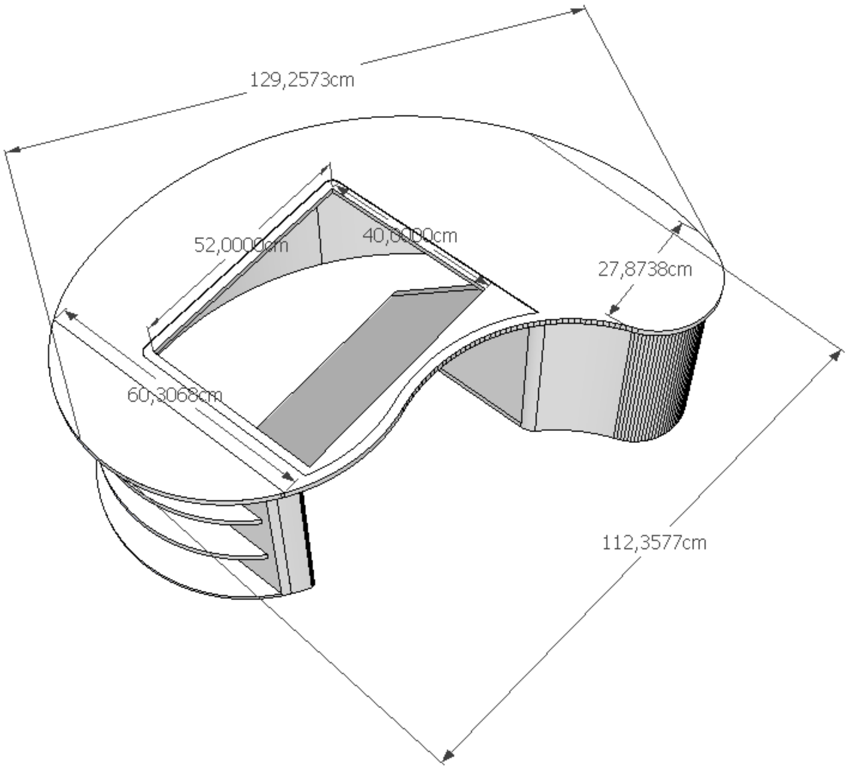


Figure 29: Interactive table top dimensions

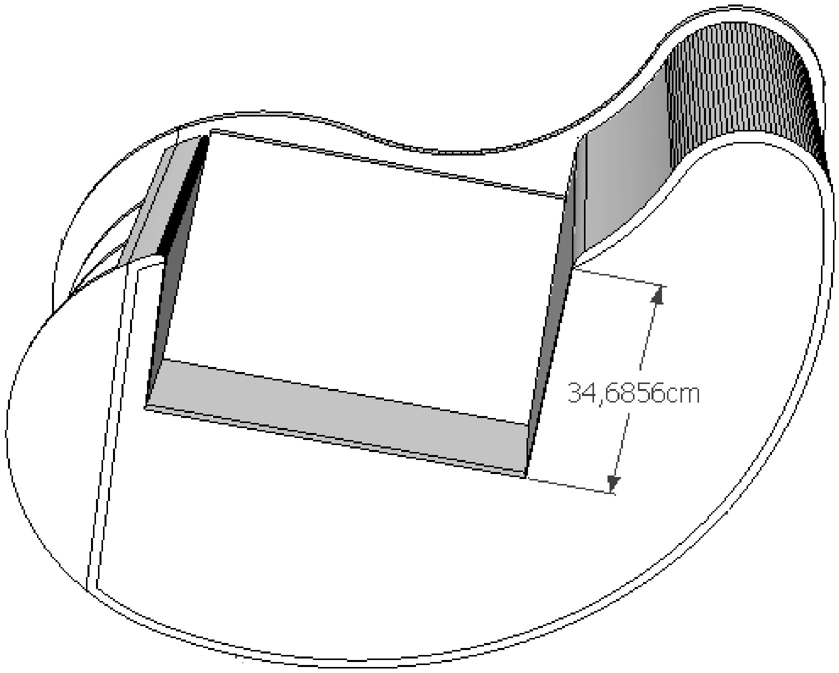


Figure 30: interactive table: space available for feet

3.2.1.2 Augmented chair

In the context of the developed framework, the utilization of monitoring and implicit input methods is of the utmost importance. The augmented interactive table is accompanied by a custom chair (see Figure 25 (2) and Figure 31), able to capture body posture, data captured by some force-pressure sensors installed beneath the seat, and a microcontroller with the capability to transmit the captured data wirelessly to the environment. Both the table and the chair have a soft yellow color which according to pedagogical experts is neutral and easily recognizable by children. The seat should not be too heavy, so that young children can easily move it around.

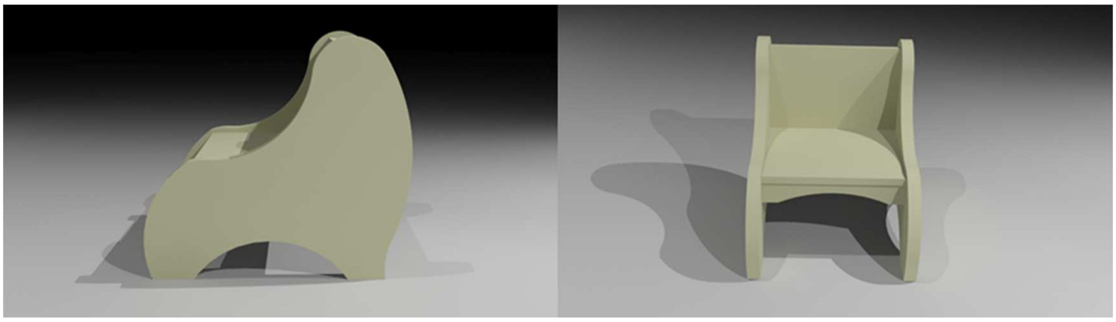


Figure 31: Custom chair

That interactive chair can be used as an alternative medium for communication and multimodal interaction. Firstly, it can act an alternative joystick, and secondly, it can be used to identify the psychological and physiological state of the children through monitoring of somatic and behavioral components.

3.2.1.3 Secondary display device

Nowadays, children are able to interact with digital content using their mobile devices almost everywhere and anytime due to the increased power, portability, and ubiquitous connectivity of mobile devices. To this end, on the top side of the interactive table (see Figure 25 (5) and Figure 32), there is a secondary display device presenting an animated model acting as a virtual child's partner during playing. The virtual character implements body animation, lip-synching, and shows emotions through facial expressions and voice tone variations. Therefore, cooperative play can be emulated by engaging the virtual partner. The remotely-controlled avatar can act as a guide, assistant or information presenter for novel, cross-platform Ambient Intelligence (AmI) edutainment scenarios. In detail, the avatar's role depends on the requirements of the AmI client applications as these are propagated remotely (using remote procedure calls). Examples of remote function invocations include real-time 3D biped skinned animations, text-to-speech, producing facial expressions and presenting multimedia content.

Figure 32: The secondary display device displaying the virtual partner

As previously referred in 2.4.3.1, the Vigotskian theory of *Zone of Proximal Development* (ZPD) and the concept of *scaffolding* [230] describe the area between what a child can do without help and the child's level of supported performance. Skills contained within a child's ZPD are the ones that have not yet occurred, but could appear if the child engaged in interactions with more capable peers or adults and/or in other supportive contexts. When learning new skills, for most children, the transition from assisted to independent performance is a gradual process that involves moving from using a great amount of support to slowly taking over until eventually no help is needed. Successful instruction within the child's ZPD involves making sure that the child will eventually be able to function independently at the same high level at which he or she was previously able to function with assistance. As result, in order to support young children skills and abilities, the remotely-controlled avatar supports child's playing by giving the appropriate amounts of support at proper times.

3.2.1.4 Sensor module

A motion sensing input device mounted on a casual wall shelf unit enables the system to provide novel interaction metaphors and techniques in the context of AmI scenarios by capturing data useful for: a) gesture/posture recognition, b) head pose estimation, c) face tracking, d) skeleton tracking, e) speech recognition, etc.

Such motion sensing device could be either Microsoft Kinect or Asus Xtion. Kinect[41] is a motion sensing input device that can bring immersive, interactive experiences to computing through voice, movement, and gesture recognition technology. The Kinect SDK enables developers to create applications by providing them with APIs, device interfaces, technical documentation and source code samples. Similar APIs are provided with the Asus Xtion.

Due to space limitations of the sensor module, it is necessary to be deployed 1.5 meter far away from the area of interest which includes the surface as well as the upper body of the child. Therefore, the sensor module is mounted on a common wall shelf unit (see Figure 33).

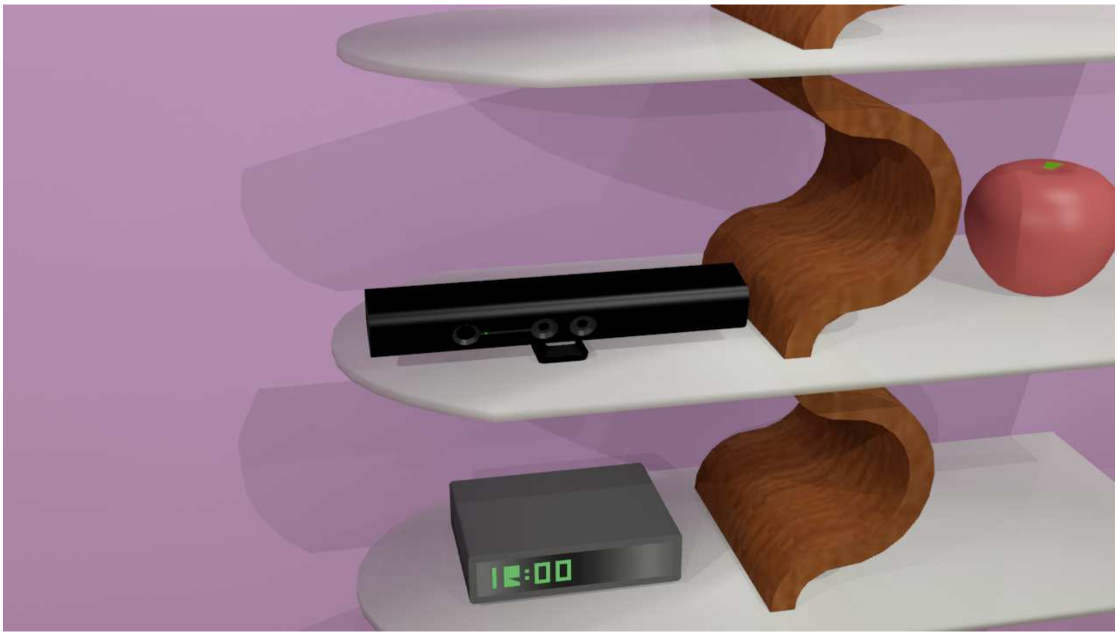


Figure 33: The sensor module may be mounted on a casual wall shelf unit

3.2.1.5 Augmented tangible objects

Ubiquitous computing technologies allow embedding computing capabilities into everyday objects and environments. Integrating technology and its use into tasks creates a number of challenges and opportunities. The design and development of the tangible objects described in the next sections relies on two main considerations:

[41] http://www.microsoft.com/en-us/kinectforwindows/

- Children learn better while playing and exploring in the physical world [170]
- Physical activity helps to build representational mappings [180].

3.2.1.5.1 Augmented toys

This section addresses the issue of the augmentation of physical toys that can be found in common games suitable for children, such as pawns, dices, cards, etc.

Figure 34: Various toys to be augmented

The physical object recognition is based on reacTIVision, an open source, cross-platform computer vision framework. Each physical object has to carry at least one fiducial marker on its abutting side, as depicted in Figure 35.

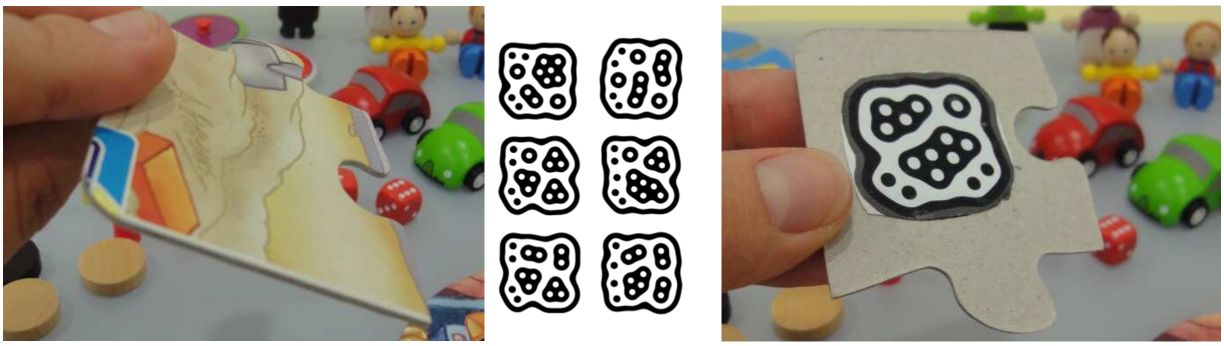


Figure 35: Augmenting a jigsaw puzzle piece using fiducial markers

ReacTIVision tracks specially designed fiducial markers in a real time video stream. The fiducial design allows the efficient calculation of the marker's center point as well as its orientation. OSC messages implementing the tangible user interface objects (TUIO) protocol encode the fiducials' presence, location, orientation and identity and transmit this data to the client applications. Additionally, reacTIVision can be used also for multi-touch finger tracking occurring at the translucent surface of the augmented table's main display.

3.2.1.5.2 Digital dice

Several traditional games such as board games consist of various types of dice (e.g., with dots, numbers or colored sides) as shown in Figure 36. Playing dice games is a good way for children to learn game skills like taking turns, staying on task, mentally adding numbers, observing others' game, etc. As a result, it is important for physical dices to be augmented in order for the system to be able to recognize their motions and identify their top side. This approach allows children to continue using typical dices in the traditional way, while the results, when the dices are thrown, can be automatically retrieved and forwarded to the system.

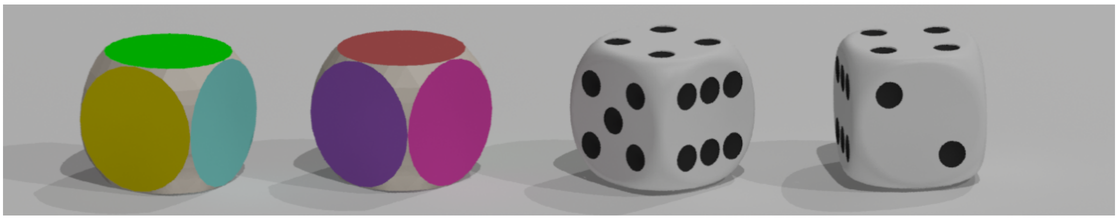


Figure 36: Various types of dices (left: color dices, right: dices with dots)

Two approaches were considered for building an augmented dice:

- A visual approach using, for example, a video camera to capture and analyze the results shown on the dice sides.
- An embedded microelectronics approach that employs a kind of integrated sensor (e.g., an accelerometer, or force pressure sensors) that internally detects the orientation of the die.

The main difference between those approaches is the size of the rolling area which in the case of the visual approach is limited to the captured image of cameras. However, in the case of the internal sensor approach, the dice's size increases dramatically, and they might need frequent maintenance such as changing the batteries. Considering the pros and cons, the embedded microelectronics approach was chosen over the visual one as further described in 4.5.5.

3.2.1.5.3 Augmented pen

Children's drawing is a way to improve physical, social, emotional, cognitive development and at the same time a way to have a lot of fun. The manipulation of a pencil reveals numerous indications of the maturity level of the children's writing skills. As a result, the augmentation of a pen is considered very important in order to measure the applied pressure, the position and the orientation of the pen on the writing surface. The augmented pen allows developers to create new and innovative pen-centric user interfaces and learn how users are affected by them.

To this end, a normal pen was turned into a transparent mediator transmitting wirelessly data relevant with the applied pressure, the position and the orientation of the pen on the writing surface. An internal sensor approach that employs some kind of integrated microelectronics (e.g. an accelerometer, or force pressure sensors) was adopted, resulting in the design shown in Figure 37 and Figure 38.

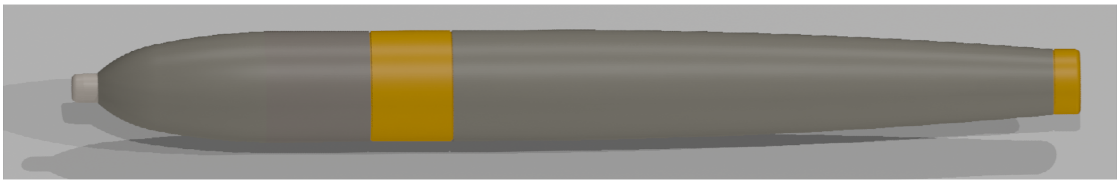


Figure 37: The augmented pen

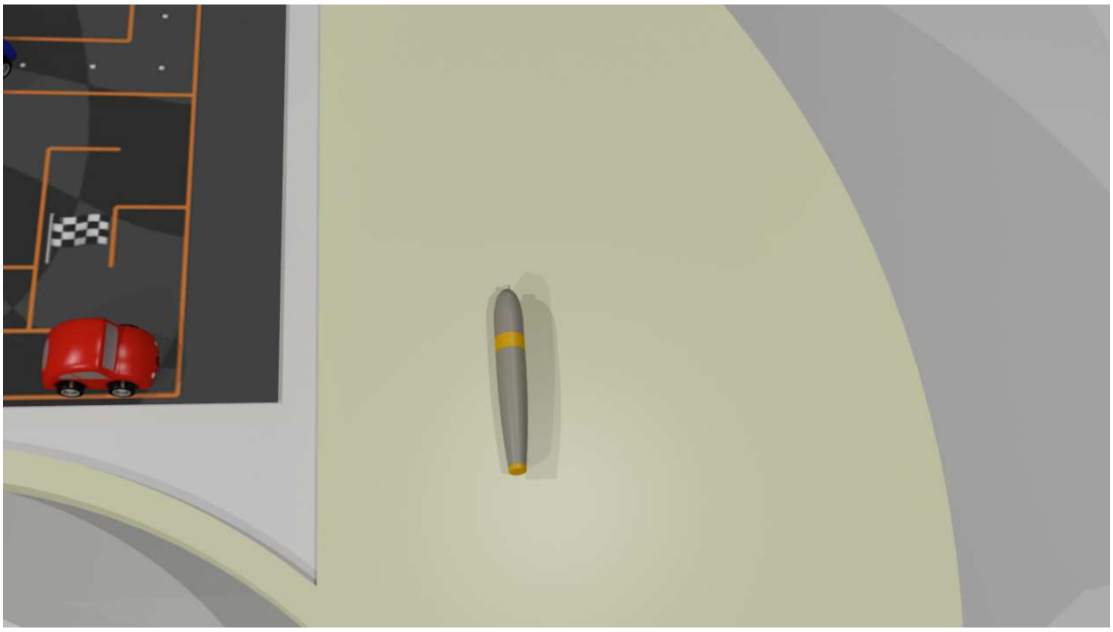


Figure 38: The augmented pen accompanies the augmented interactive table

In order to recognize the position of the pen, an infrared led has been installed in the pen's tip. This infrared led lights automatically when pressure is applied to the pen, and consequently the vision software components is able to identify a blob on the writing surface.

3.2.2 Activity analysis and play performance

As mentioned in the Background section, play is invaluable in assessing children's development and providing intervention, and is one of the areas of human occupation that occupational therapy focuses on. Early intervention professionals focus on monitoring appropriate activities that children do in order to explore and facilitate their skills and abilities. **Activity analysis** is an important process carried out by occupational therapists in order to understand the various demands that a specific activity requires for execution and completion.

The proposed framework aims to facilitate the **Activity analysis** process through the provision of appropriate **AmI applications** (i.e., tools**,** libraries, etc.), and the easy integration of AmI technologies into children's playtime. As a result, children are provided with interactive games that are constantly adapted to their unique and continuously changing characteristics, and at the same time can act as assessment tools for early intervention professionals, providing relevant information about each activity occurred during playing. Moreover, parents can any time be informed about their children's development progress. The system, apart from the initial knowledge about children's characteristics, continuously gathers and stores information about children's skills and abilities (i.e., interaction data) through the use of **smart games** in a predetermined protocol. The system analyzes this set of data and makes the results available back to the smart games or exposes them to parents or early intervention professionals.

There are specific behaviors and skills expected from children as typical or normal according to their age. Using standardized tests and measures provides an excellent foundation for assessing child development for early intervention professionals. Many standardized assessment tools and measures have been developed to assist this process. Most developmental and functional tests assess a child's activity and participation, although some tests have items to evaluate body structures and functions as well. The International classification of functioning, disability and health, Children & Youth version (ICF-CY) [245], offers a conceptual framework and a common language and terminology for those who want to document characteristics of health and functioning in children and youth. ICF-CY expands the coverage of the ICF (WHO 2001) to encompass the body functions and structures, activities, participation and environments specific from birth to age 18 years.

For the purposes of this thesis, the Denver II, a revision of the Denver Developmental Screening Test (DDST), commonly known as the Denver Scale [69] has been selected. Denver II is a widely accepted pediatric screening tool for ages up to 6 years old.

The process of the **Activity analysis** contains the identification of the tasks which are involved during young children's play. Task monitoring (when applicable) may result into the measurement of the play performance and capacity which, in comparison with Denver II scale's expected scores, can drive to the extraction of useful indications about young children's development.

Play performance is a measurement concept that can describe what tasks a child does while playing in a specific context and environment [238]. **Capacity** is a measurement concept that indicates the highest probable level of functioning that a child may reach in the ICF-CY domain of Activities and Participation at a certain moment [238]. Capacity is measured in a standard environment, in order to reflect the environmentally adjusted ability of the individual. In summary, a capacity qualifier specifies what a child can do independent of context, i.e. the best ability to execute a task or an action in a standardized environment, whereas the performance qualifier specifies what a child does do in a current environment, i.e. functional skills used in everyday life situations [146].

In order to analyze play performance, a systematic process is needed to address the factors that may affect child's functionality and identify the context related factors, as well as the interrelations among them. All the factors are interrelated and can be grouped according to their nature and origin as: (a) **child factors**, (b) **performance skills**, (c) **activity demands** and (d) **context and environment**.

Child factors are factors related to individual or population characteristics that may affect performance while playing. According to ICF-CY, **Body Functions** are physiological functions of body systems (including psychological functions) and Body Structures are anatomical parts of the body such as organs, limbs and their components (WHO 2007). Numerous body functions and structures underlie and enable child's performance. Any deviation of what considered normal, loss or influence of an environmental factor in a body function or structure can affect child's functionality and execution of several related tasks.

Performance skills are observable, concrete, goal directed actions that children use to engage in play situations [64]. From an occupational therapy perspective performance skills are defined as the abilities children demonstrate in the actions they perform. They include:

- **Sensory – Perceptual skills** are the actions or behaviors that a child uses to locate, identify and respond to visual, auditory, proprioceptive, vestibular and tactile stimuli
- **Motor skills** are planning, sequencing and executing movements.
- **Praxis skills** can simply defined as "what to do and how to do it" and can be observed when a child is imitating, constructing or sequencing actions or behaviors. Praxis is the neurological process by which cognition directs motor action (Ayres, 1985).
- **Cognitive skills** are actions and behaviors that a child uses to plan and manage the performance of game activities
- **Communication and Social skills** are what purposefully child does for interacting with others. Actions or behaviors such as eye contact, gesturing, dialogue, taking turns, recognizing others perspectives, intentions and actions are included.
- **Emotional Regulation skills** are actions or behaviors a child uses for managing and expressing feelings during activities or interaction with others.

Activity demands are the aspects of an activity, including the objects and their properties, required actions and skills, space and social demands, required or underlying body functions, structures, timing and sequencing needed to carry out the activity (play a game).

Context and environment consist of a verity of interrelated conditions that may influence child's performance. Environmental conditions are relating the external physical and social environment in which play occurs. **Context** refers to cultural, physical, social, spiritual, temporal and **virtual circumstances** during play.

3.2.3 Smart games

ICT technology has provided the means to produce games that employ the environment to offer novel play experiences to children. However, little work has been conducted so far to facilitate such technology to offer games that focus on play and concurrently are valuable for monitoring child development. Currently, such monitoring is practiced for young children mainly through physical games. In that case, occupational therapists observe young children playing in order to capture measurements about their development progress by using some of the aforementioned available assessment tools.

The proposed framework aims at facilitating the implementation process of games by simplifying the coexistence and the cooperation among physical and digital objects, virtual avatars and an adaptation infrastructure mechanism. Traditional games are turned into **smart games** that can: a) enhance playing experience, b) monitor children's changing abilities and consequently adapt to them, and c) act as assessment tools for early intervention professionals providing them with relevant information (i.e., play performance) about each activity occurred during playing.

Early intervention professionals have a primary role during the smart game design process. Each smart game is analyzed into its individual characteristics. In general, game's characteristics fall into the following categories:

- **Content:** includes physical or digital objects
- **Actions:** include instructions and appropriate guidance
- **Aim of game:** includes success criteria to complete the game, such as requested help, playing duration, content understanding and potential mistakes in the execution of the game, which may not be identified or corrected by the child during playing
- **Required activities**: Required or targeted abilities based on the ICF-CY.

It is important to take into consideration that environmental stimuli might influence child's play performance. To this end, all relevant context settings should be addressed, recognized and elaborated when analyzing playing demands.

The following table shows the factors included to support the design rationale of each smart game.

Game	<name of the game>
Play/Activity demands	Equipment and materials, devices and properties/characteristics
Context demands	Physical and Virtual environment (how communication/ interaction with the system occurs)
Social demands	Rules of the game
Grading Demands	
Sequencing and timing	Game description (process to complete the game: steps/states, sequence, time requirements)

Required actions / Performance skills (ICF-CY Activity and Participation)	The performance skills that would be used from anyone to play the game (e.g. action/task > pick a card – handling objects). Skills are categorized in sensory, perceptual, motor, praxis, cognitive, communication and social.
Body functions (ICF-CY). Physiological and psychological functions that are required to execute the actions for playing the game.	Global Mental functions
	Specific mental functions (Attention, Perception, Thought, Higher level cognitive functions, mental functions for sequencing complex movement, experience of self and time)
	Sensory functions (seeing and related functions, hearing functions, vestibular functions, proprioceptive functions and touch)
	Movement related functions (control of voluntary movement, joint stability?)
	Voice and speech functions
Safety demands	(if any)

Table 8: Smart game design rationale

Each smart game has various levels of difficulty. There are many different activities required from children at the various game levels. According to Denver II's scale, each game level is designed to focus on exclusively one primary (or specific) fully developed ability of a child. Additionally, a variety of general activities can be involved during play. The activity analysis process contributes to the identification and examination of each required activity during play, either specific or general. The main idea is that during play, the smart game monitors and evaluates the play performance and to commit a representative score to the adaptation infrastructure mechanism. The latter analyzes the play performance of the current level's specific activity and makes appropriate adaptation suggestions back to the smart game.

3.2.4 Reporting child's performance

The proposed framework extracts, through interaction monitoring, indications of the achieved maturity level and skills of the child. Furthermore, it provides reports to parents and early intervention professionals. These reports contain information about the child's capacity to execute the required activities involved during game playing. When a child performs a game in a uniform or standard environment, his capacity can be used as an indication of the child's real ability in required activities of the game. Low capacity scores may be an indication of a functional difficulty that a child may have in a given developmental area. Capacity indicators are reported according to: a) data collected at certain points in time, b) child's age, and c) degree of independent completion of the game or some specific tasks (with or without system's assistance or guidance). Data should be collected from the first time (session) that the child has played a game and repeatedly after some period of time (defined from the system administrator), i.e., a month/ year or after X sessions.

Two types of report are supported, a basic one suitable for parents and an extended one suitable for early intervention professionals. Independently from the type of the report, the following table shows the common contents.

Game name, date, time
Child's profile characteristics including impairments or possible delayed skills and abilities.
System factors (information of the environmental factors in which the data has been collected)
History of interaction (Number of session, total number of sessions for the smart game, etc.)
Overall capacity according to age standards and the completion of the predefined levels (measured of playing without help) (graphical representation, scale)
Level in game (graphical representation, percentage scale)
Explanation of the graphical representation. "(Name of the child) experienced: • no difficulty in (name of tasks that the score is between 95-100 %) • slight/mild difficulty in (name of tasks that the score is between 95-75 %) • moderate/medium difficulty in (name of tasks that the score is between 50-75 %) • high/severe difficulty in (name of tasks that the score is between 0-50 %)
Recommendations area

Table 9: Contents of reports either for parents or early intervention professionals

Regarding the recommendation area, in case the child reaches a score value between 0-5% (which means complete inability) the following message appears:

"(Name of the child) has unsuccessfully tried to play (name of the game) (number of times) times and a detailed observation of what causes this inability is recommended."

If there are tasks where mild - severe difficulty appears and the capacity level is below age level the following message will appear in this area:

"(Name of the child) has played (name of the game) (number of times) times and he/she appears to have some difficulties in (number of tasks) required tasks for successfully completed the game. A detailed/ more analytic observation of his /her problems by your pediatrician or professional in child's development is recommended."

Professional's report contain information regarding child's **performance** in tasks or activities during a playing session. Performance data is collected throughout a session, while they contain information about the amount and the type of system guidance and the assistance that the child has used, as well. Performance data can be used for monitoring:

- the child' s interaction with the system
- his performance capabilities in various activities
- the functional level in specific (predefined) activities according to age
- the presence of potential problems in body functions that are required to play a specific game
- child' s progress in acquiring new skills through games.

4 Framework implementation & deployment

This chapter describes the implementation of the developed framework (**Bean framework**) aiming to actively anticipate users' needs through the provision of automatic and proactive services. The framework components include sensory modules, augmented artifacts and administration facilities. The modularity provided by these components eases the cooperation among them, promotes reuse and allows adding new components as needed. This section concludes with the in vitro deployment of the novel framework in the AmI playroom (i.e., ICS-FORTH's AmI simulation space).

4.1 High-level system architecture

In the context of a distributed AmI environment, system architectures become fuzzier than in traditional systems due to the fact that services distributed in the environment have their internal architecture and structure. However, the way that services are incorporated and used forms another higher level architecture. A generic high-level architecture of the **Bean framework** is presented in Figure 39.

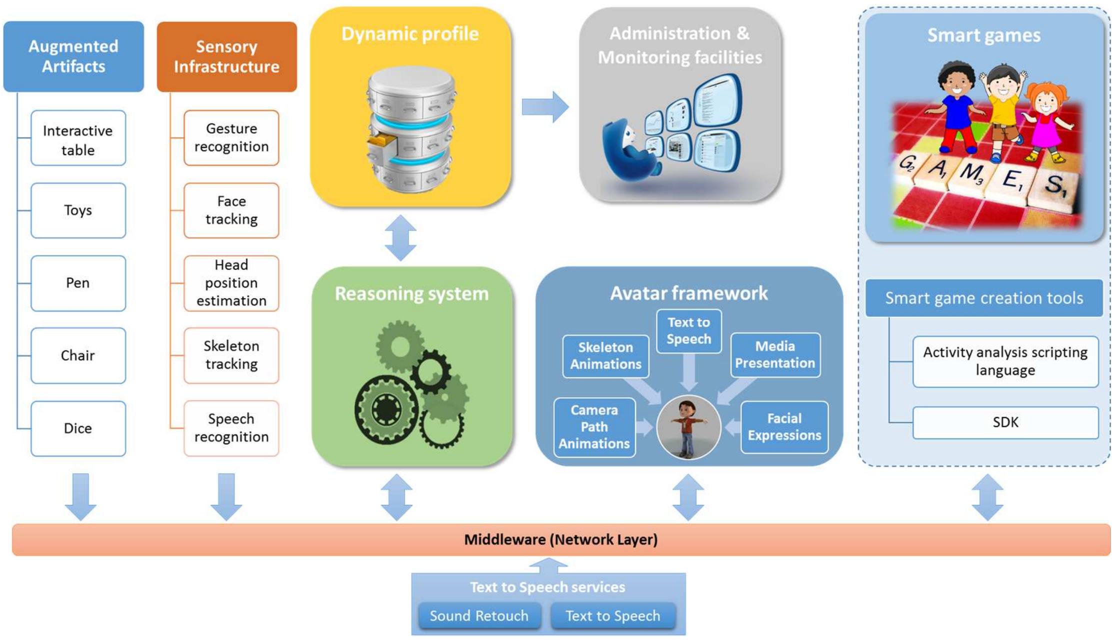


Figure 39: Framework architecture

More specifically, Figure 39 illustrates the high-level components that cooperate with one another through the middleware network layer, which consists of: a) sensory infrastructure, b) dynamic profile, c) reasoning system, d) augmented artifacts, e) administration and monitoring tools, f) avatar framework, and g) smart games. The **sensory infrastructure** module aims to simplify the creation of novel interaction techniques supporting natural and intuitive interaction of children with everyday objects. The **Dynamic profile** is an enriched dynamic profile model of information about children's skills and abilities facilitating the "learning" process of each child's specific developmental progress, preferences, habits and abilities. Additionally, the **Dynamic model** is a centralized repository of information used to describe the required activities of each smart game, the interaction history, as well as the user's roles. The **reasoning system** estimates child's play performance and therefore, it facilitates the adaptation of the smart game's interaction model and the provided services, thus encouraging the child's abilities development in an unobtrusive manner. Finally, the **Middleware (network layer)** is designed to facilitate the communication systems deployed on diverse platforms, as presented in [220]. In other words, it facilitates the communication among the aforementioned components, the envisioned **smart games, augmented artifacts** and **avatar framework**, as well as the **administration and monitoring facilities** proposed in the current thesis.

4.2 Bean model: an enriched dynamic profile model

As mentioned in the previous sections, the high level goal of the proposed framework is to monitor, evaluate and enhance children's skills and abilities through playing. To this end, an enriched dynamic profile model was designed and implemented to act as a centralized repository of information relevant to: a) children's profile and characteristics, b) smart games, c) data extracted from the activity analysis process (see section 3.2.2), d) interaction history, and e) employed users of the proposed framework.

Microsoft SQL server is the technology used to store the aforementioned complex structure. The term **"Bean model"** refers to the entity data model which is illustrated in Figure 40 and described in detail in the following paragraphs. According to the literature, an entity data model is a set of concepts that describe the structure of data, regardless of its stored form[42]. Adopting the structure of the ICF-CY, the entity **"Domain"** provides information (e.g., title) about each used ICF-CY component, such as Body Functions and Structures, Activities and Participation. Each component consists of various domains and, within each domain there are categories, which constitute the units of classification. To that end, the entity **"DomainSubCategory"** is used to describe (e.g., title) the classification units used by the framework (see Table 10).

Categories / Units of classification	ICF-CY components
Mental functions	Body functions and structure
Sensory functions	Body functions and structure
Voice and speech functions	Body functions and structure
Neuromusculoskeletal and movement-related functions	Body functions and structure
Purposeful sensory experiences	Activities and Participation
Basic learning	Activities and Participation
Applying knowledge	Activities and Participation
General tasks and demands	Activities and Participation
Communication	Activities and Participation

Table 10: Units of ICF-CY classification used by the framework

[42] https://msdn.microsoft.com/en-us/library/vstudio/ee382825%28v=vs.100%29.aspx

The subset of skills and abilities selected as the target of the proposed monitoring and evaluation functionality was the product of the conducted user requirement elicitation sessions with early intervention professionals. Such skills and abilities regard various ICF-CY activities (Activities and Participation) that a smart game may require during child's play. Each required activity involves a set of body functions (Body functions and Structure). Thus, the navigation property **"InvolvedBodyFunctions"** facilitates this structure by connecting each entry in the entity **"Activity"** to its involved body functions (i.e., entries in the entity **"BodyFunction"**).

Further properties, stemming from the ICF-CY, are used to describe in more detail the entities **"Activity"** and **"BodyFunction"** with the following characteristics: a) name, b) description and c) code. At this point, the Denver II's scale is imported to the model in order to provide relevant information about each activity's expected capacity through the navigation property **"ExpectedCapacityByAges".** The latter establishes a one-to-many relationship between the entity **"Activity"** and the entity **"ExpectedCapacityByAge".** Furthermore, the entity **"ExpectedCapacityByAge"** contains Denver II's scale data (i.e. Age, Capacity) that reflects the percentage of a certain age group able to perform the connected activity. The following table depicts a representative sample of the initial subset (for the purposes of this thesis) of activities and their involved body functions.

ICF-CY category \ Activities		**Expected capacity**				
		3y	**4y**	**5y**	**6y**	**Involved Body Functions**
Applying knowledge	**d1601**, Focusing attention to changes in the environment	25	50	75	100	**b1400** Sustaining attention
						b1401 Shifting attention
	d161, Directing attention	25	50	75	100	**b140** Attention functions
						b1400 Sustaining attention
						b1401 Shifting attention
						b1402 Dividing attention
						b1403 Sharing attention
	d1630, …	25	50	75	100	**b1603** Control of though
						…
Basic Learning	**d130**, Copying	25	50	75	100	**b147** Psychomotor functions
						b163 Basic cognitive functions
						b760 Control of voluntary movement functions
	d131, Learning through actions with objects	25	50	75	100	**b1565** Visuospatial perception
						b163 Basic cognitive functions
						…
	…					…
⋮	…					…

Table 11: An example of selected activities, their expected capacity per age and their involved body functions.

In the context of interaction monitoring, it is essential to model human tasks that a child may execute during playing as reactions to system output. The entity **"HumanTask"** is used to store all the envisioned human tasks presented in the Table 12.

Matching a physical object with the corresponding virtual object
Place a number of physical objects on virtual places
Respond to simple command
Follow simple direction or instruction
Follow complex directions or instructions
Listening to sound or speech produced by the system
Viewing, Seeing, Observing or tracking animation or video events produced from the system
Touch, press, put, place or lift physical or virtual objects on the interface
Standing, sitting and shifting body; center of gravity
Remaining sited or standing or in same posture for a required period of time
Imitation of system demonstration(s)
Interacting with virtual objects
Playing with physical objects
Attention span intentionally focusing on specific stimuli, by filtering out distractions.
Intentionally focusing in system's output (stimuli) during interaction
Imitation of postures demonstrated by a three-dimensional virtual character
Picking up, grasping, manipulating and releasing physical objects
Locate and looking at picture(s), figure(s) or image(s) presented on the interface
Select, manipulate and place a physical object on the matching virtual object
Preparing, initiating, completing and arranging the time for playing a game without assistance
Comprehending and properly responding to the meaning messages conveyed by gestures, general signs and symbols drawings and photographs
Receiving and comprehending messages in formal sign language with literal and implied meaning
Comprehending the literal and implied meanings of written messages, including Braille
Brink game box or empty or collect physical objects
Respond to system prompt timely
Choose the required physical objects, initiating the game session and arranging the time for playing a game

Table 12: Modeled human tasks monitored during interaction

In the same context, the entity **"SystemTask"** is used to describe system tasks regarding the employed communication functionality (both for input and output) between the user and the system. Through the proposed augmentation of everyday objects and the implementation of a novel sensory infrastructure, such system tasks were envisioned and modeled as depicted in the following table.

Gesture/Posture recognition - (system input)
Head pose estimation using - (system input)
Face tracking - (system input)
Skeleton tracking - (system input)

Speech recognition - (system input)
Physical object recognition - (system input)
Cursor recognition (Multitouch) - (system input)
Force pressure recognition at the interaction surface - (system input)
Force pressure recognition while sitting - (system input)
Force pressure recognition at the pen's tip - (system input)
Motion sensing cube (e.g. dice) - (system input)
Presentation of still images - (system output)
Presentation of audio - (system output)
Presentation of text - (system output)
Presentation of animation - (system output)
Presentation of video - (system output)
Presentation of speech - (system output)
Sign language - (system output)
Braille; tactile writing system - (system output)
Other assistive devices for people with hearing, voice, speech, or language disorders
Other assistive technologies for people with visual disorders

Table 13: Modeled system tasks interfacing user for both input and output

As mentioned in section 3.2.2, a smart game is able to monitor and evaluate the play performance for selected **required activities**. These activities are divided into **general** and **specific**, and are modeled into the database entity **"RequiredActivity".** The latter contains navigation properties to a) the corresponding smart game (e.g. entity **"Application"**), b) the actual ICF-CY activity, c) a subset of the activity's associated human tasks that may carry further description to define more accurately the user interactions during playing (with the corresponding smart game) and d) a set of the employed system tasks required for the implementation of a certain smart game.

To begin with, as illustrated in Figure 40, the navigation property **"HumanTaskExts"** establishes a one-to-many relationship between the entity **"RequiredActivity"** and the entity **"HumanTaskExt".** The entity **"HumanTaskExt"** stores a detailed description of a required activity's human task that may occur during playing with a selected game. Similarly, the navigation property **"SystemTaskExts"** establishes a one-to-many relationship between the entity **"RequiredActivity"** and the entity **"SystemTaskExt".** The latter stores a detailed description of how the system interfacing users (i.e., children) in both input and output in the context of a particular required activity yet again during playing with a selected game.

For example, a smart game called "**Game#1"** requires the activity ***"Acquiring basic concepts"*** (d1370). This has as a result the creation of an entry in the entity **"RequiredActivity"** which has a navigation property to the corresponding entry in the entity **"Activity"**). Regardless of the selected game, this activity was already linked with three human tasks including the ***"Matching a physical object with the corresponding virtual object"*** (i.e., entries in the entity **"HumanTask"**). However, the selected game "**Game#1"** encourages and expects the child to execute only one of the total three linked human tasks. To this end, a unique entry is created in the entity **"HumanTaskExt"** in order to establish a link between a certain human task and the required activity. At first, that entry contains a copy of the original human task description. However, in the case of **"Game#1"**, that description has been slightly changed reflecting the task that a child may execute during playing, namely ***"positioning of wooden animals on their corresponding virtual positions"***.

Similarly, the activity has already been linked with various system tasks including the ***"Physical object recognition"*** *task*. According to the gameplay of **"Game#1"**, physical objects such as puzzle pieces are manipulated by the child. Therefore, an entry is created in the entity **"SystemTaskExt"** which establishes a connection between the required activity and necessary system input functionality. This entry initially contains a copy of the original system task description that can be changed as in the case of the human task mentioned before.

The entity **"RequiredActivity"**, beyond the already specified navigation properties, contains descriptive properties such as a) "**IsGeneral**", and b) "**IsSpecific".** The primitive data type of such properties is string in order to contain information about the age range related to various levels of the smart game. For instance, in **"Game#1"** the required activity ***"Acquiring basic concepts"*** (d1370) is specifically monitored in the 3rd level of the game by setting the **"IsSpecific"** value to **"6-7|"**. In this case, the value of the property **"IsGeneral"** is **"|"**, meaning that is not *general.* The latter corresponds to the smart game design rationale which states that a required activity cannot be set as *specific* and *general* at the same time. Another example concerns the gameplay of **"Game#1"**, based on the fact that the ability ***"Listening"*** (d115) is fully matured in children of the age range 3 to 6, according to the Denver II's scale. Thus, this ability constitutes a required activity for the selected game and has been set as *general,* implying that it is a prerequisite for all available game levels. Concluding, the values of the corresponding properties are: a) **"IsGeneral"** = "3-4|4-5|5-6|6-7", and b) "**IsSpecific"** = "|".

The entity **"PerformaceRecord"** is used to describe the monitoring results during playing. The latter is responsible to estimate the child's play performance in each required activities according to the current gameplay. In detail, each entry in the entity corresponds to an estimated score of child's play performance for a selected required activity within the range 0-100. Other properties are a) **"DateTime"**: the timestamp when the play performance was estimated, b) **"Comments"**: any textual representation of the context, and c) **"Footage"**: any unedited visual or audio material captured during the estimation process or raw data captured from the sensors of the envisioned sensory infrastructure. Beyond descriptive properties, there are navigation properties such as a) the required activity which receives the play performance score (a many-to-one relationship between the entity **"PerformanceRecord"** and the entity **"RequiredActivity"**), and b) the session which is a period of time within which the performance record was committed (a many-to-one relationship between the entity **"PerformaceRecord"** and the entity **"Session"**).

The entity **"Session"** models the period of time during which the user (i.e., child) plays a selected smart game. Its fundamental properties are: a) "**StartDateTime**": the time when the child selected a smart game to play with, and b) "**EndDateTime**": the time when the child quits the selected smart game or the time when interaction stopped by itself due to idle timeout. Beyond these properties, the entity has the following navigation properties: a) the smart game (i.e. **"Application"**) of which interaction started (a many-to-one relationship between the entity **"Session"** and the entity **"Application"**), b) the child (i.e., **"User"**) who selected to play (a many-to-one relationship between the entity **"Session"** and the entity **"User"**) and c) a list of committed performance records (i.e., **"PerformanceRecords"**) occurred during playing with the selected game (a one-to-many relationship between the entity **"Session"** and the entity **"PerformaceRecord"**).

As depicted in Figure 40, the entity **"User",** represents the basic user profile characteristics via the usage of the following properties: a) **"Name"**, b) **"Surname"**, c) **"BirthDate"**, d) **"Gender"**, e) **"Language"**: native spoken language and f) **"Tag"**: a tag id that can be used for user identification by employing common technologies such as RFID tags or QR codes. Furthermore, the same entity has properties about the user's role in the proposed framework, such as a) "**IsAdministrator**" which returns true in case that the user has administration privileges, b) "**IsChildDevelopmentExpert**" which returns true in case of user is child care professional, and c) "**IsParent**" which accordingly returns true in case the user is parent. If all the aforementioned properties return false, then the underlying entry belongs to a child. The following two properties were added to the entity in order to facilitate basic authentication functionality for all users except children: a) "**Username"**, and b) "**Password"**.

Moreover, the entity **"User"** has a few navigation properties such as: a) **"Sessions"** which refers to a list of entries in the entity **"Session"** (a one-to-many relationship between the entity **"User"** and the entity **"Session"**), b) **"ActivityLimitations"** which refers to a list of entries in the entity "**Activity"** in which the child has limitation (i.e. capacity equals to "0") in their execution (a many-to-many relationship between the entity **"User"** and the entity **"Activity"**), c) **"BodyFunctionLimitations"** which refers to a list of entries in the entity "**BodyFunction"** in which the child has functional problems (a many-to-many relationship between the entity **"User"** and the entity **"BodyFunction"**). Moreover, the entity has a navigation property to its **"AssociatedUsers"** (a many-to-many relationship between the entity **"User"** and the entity **"User"**), facilitating the connection between parents and their children, as well as between the early intervention professionals with their clients.

The last entity **"Application"** is used to model the basic characteristics of a smart game through the following properties: a) **"Name"** which refers to the game's title, and b) **"Description"** which is a textual description of the gameplay or goals. Further than these properties, an entry of the entity **"Application"** contains information about a smart game such as required activities or sessions created so far. To this end, the entity has the following navigation properties: game a) **"Sessions"** (a one-to-many relationship between the entity **"Application"** and the entity **"Session"**) which refers to a list of entries in the entity **"Session"** which created during playing with the selected smart game, and b) **"RequiredActivities"** (a one-to-many relationship between the entity **"Application"** and the entity **"RequiredActivity"**) which is a list of activities required for playing with the selected smart game.

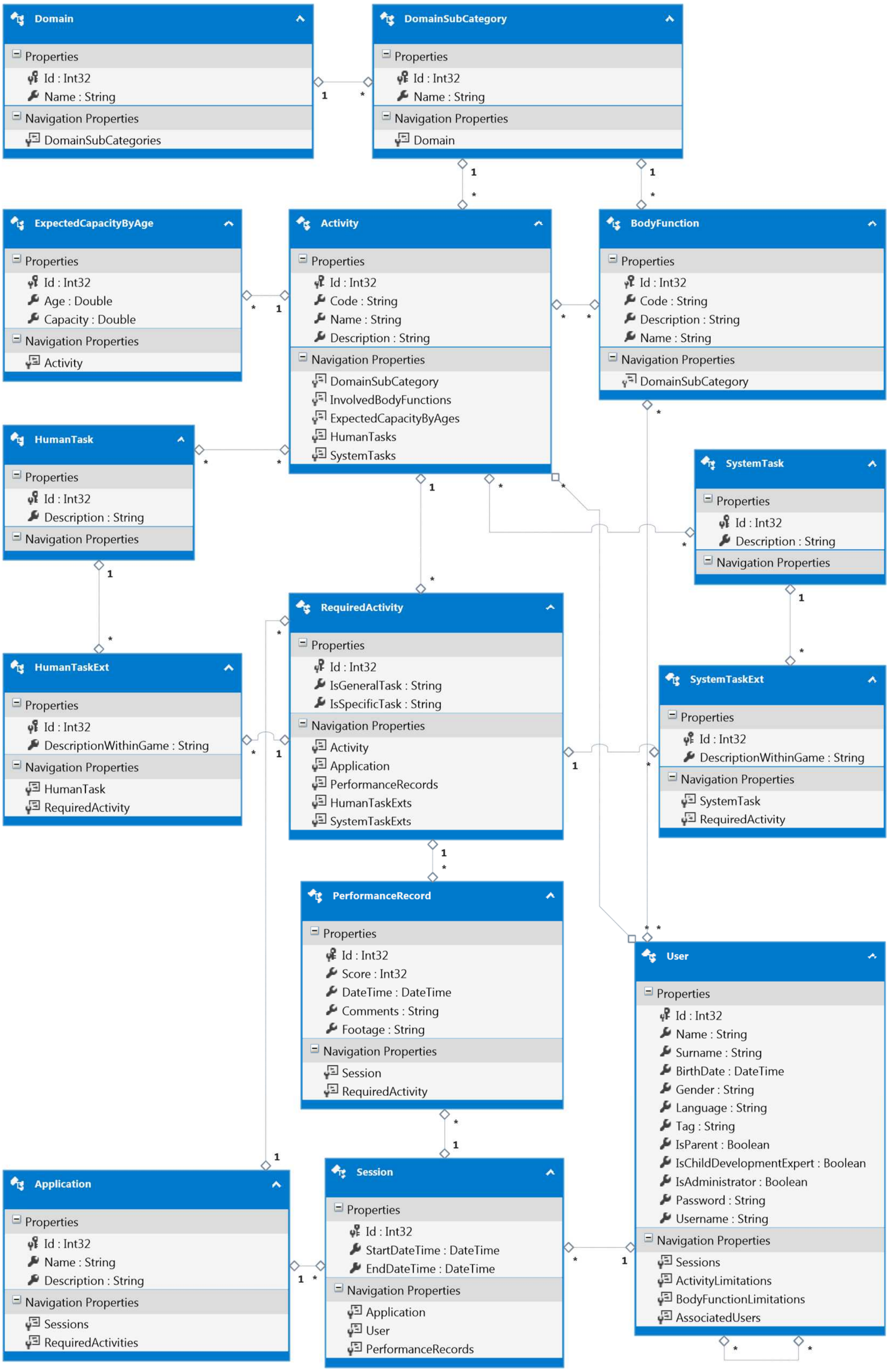


Figure 40: Bean model (Entity data model)

4.3 ADAM: An **ADA**ptation infrastructure **M**echanism

The adaptation of each smart game is facilitated through the adaptation infrastructure mechanism (ADAM). ADAM is built upon the Entity Framework to access the existing database (i.e., **Bean model)**. Entity Framework (EF) is an object-relational mapper that enables .NET developers to work with relational data using domain-specific objects.

During play, the selected smart game is responsible to monitor and evaluate child's play performance and commit a representative score (range 0-100) to ADAM. The latter monitors and evaluates child's maturity at the various levels of a selected smart game. In detail, it collects and analyzes the child's play performance commitments regarding *specific* and *general* activities required for the active game level and makes appropriate adaptation suggestions back to the smart game. The analysis is conducted using time series forecasting methods (i.e., weighted moving average). Using this analysis, the recorded data are imported to the time series in order to generate the developmental curve of the targeted specific activity of the currently active game level.

In the weighted moving average model, every play performance value is weighted with a factor from a weighting group. Thus, recent historical data (play performance commitments) have greater influence when determining the capacity of a selected required activity. This approach was chosen because more recent play performance data are more representative and reliable than older data. Therefore, ADAM is able to react more quickly to a change in the play performance during playing. The accuracy of this methodology depends largely on the choice of the weighting factors. With the help of early intervention professionals, the selected factors are W_{15}=15, W_{14}=14… W_1=1. To this end, the capacity of a required activity (AC) is the product of the following formula

$$AC = \frac{\sum_{t=1}^{n} W_t \times V_t}{\sum_{t=1}^{n} W_t}$$

, where W_t is the weighting factor, V_t is the value of the play performance, n is the number of the weighting factors (n=15) and AC is the average value representing the child's capacity to execute a required activity.

As a result, through statistical analysis the adaptation infrastructure mechanism extracts the current child's capacity in the execution of various activities and estimates the developmental rate based on the entire interaction history (i.e., past play performance commitments). Therefore, the adaptation infrastructure mechanism is able to identify children whose play performance deviates significantly from the average population of their age. At the same time, this implies that further investigation is recommended to determine if there are any developmental problems that require special care. To this end, a smart game can adapt to the child's evolving skills so as to choose the most appropriate level according to child's estimated abilities.

4.3.1 Using ADAM from a developer's perspective

As mentioned earlier, ADAM remotely provides smart games with the child's profile and adaptation suggestions by employing a service oriented middleware network layer. The latter is designed to facilitate the communication of systems that are deployed on multiple platforms as presented in [76]. In detail, it provides smart games with child's basic information such as name, surname, birthdate, etc., as well as information related to functions and structures of the body, activity limitations and participation restrictions.

ADAM implements the following IDL[43] interface (see CodeBlock 1) which is mainly divided into three categories: a) User information, b) Session management, c) Play performance and capacity information, and d) Adaptation suggestions – notifications.

```
struct UserInfo
{
       string    name;
       string    surname;
       long long birthdate;
       string    gender;
       string    language;
       boolean   isParent;
       boolean   isChildDevelopmentExpert;
       long      tag;
};

///////////////////
// User information
///////////////////

boolean LookupUser(in long user, out UserInfo userInfo, out string error);

/////////////////////
// Session Management
/////////////////////

boolean StartSession(in long user, in string game, out long sessionId, out
string error);

boolean CommitPerformanceRecord(in long sessionId, in string activity, in
string scope, in long score, in string comments, in string footage, out
string error);

boolean EndSession(in long sessionId, out string error);

/////////////////////////////////////////////
// Play performance and capacity information
/////////////////////////////////////////////

boolean CalculateUserCapacityInActivity (in long user, in string activity,
out double capacity, out string error);

boolean CalculateUserCapacityInActivityInGame (in long user, in string
activity, in string game, out double capacity, out string error);

boolean CalculateUserCapacityInActivityWithScopeInGame (in long user, in
string activity, in string scope, in string game, out double capacity, out
string error);

boolean HasUserLimitationInActivity(in long user, in string activity, out
```

[43] http://en.wikipedia.org/wiki/Interface_description_language

```
string error);

boolean HasUserLimitationOfBodyFunction(in long user, in string
bodyFunction, out string error);

boolean IsUserCapable(in long user, in ami::StringSeq activities, in
ami::StringSeq bodyFunctions, out string error);

boolean HasUserSufficientCapacityInActivity(in long user, in double
capacity, in string requiredActivity, in string game, out string error);

boolean ComputeInferredUserAgeInGame(in long user, in string game, out long
age, out string error);

/////////////////////////////////////////
// Adaptation suggestions – notifications
/////////////////////////////////////////

ami::StringSeq GetAssociatedSystemTasksForActivity(in string activity);

ami::StringSeq GetAssociatedHumanTasksForActivity(in string activity);

void Event_InferredUserAgeInGameChanged (in long user, in string game, in
long fromAge, in long toAge);

void Event_InferredActivityCapacityInGameChanged (in long user, in string
game, in string requiredActivity, in long oldValue, in long newValue);

void Event_TryAssessEquivalentActivity (in long user, in string game, in
string oldActivity, in string newActivity);

void Event_TryDifferentMultimodality (in long user, in string game, in
string selectedActivity, in string oldSystemTask, in string newSystemTask);
```

CodeBlock 1: ADAM's implemented functionality – exposed to smart games (IDL)

Firstly, the smart game requests the child's basic information by remotely calling the function **LookupUser.** The latter takes as argument the child's id or tag (i.e., value of the *user* parameter) coming from an identity card (e.g., RFID, QR Code or other special visual marker) and returns an instance of the struct *UserInfo.* This struct contains the attributes: a) *name*, b) *surname*, c) *birthdate*, d) *gender*, e) *language*, f) *isParent*, g) *isChildDevelopmentExpert*, and h) the *tag* itself. Secondly, the selected game requests ADAM to start a new session by calling the function **StartSession** in order to create a new entry in the entity Session and to return its **id** back to the smart game. The latter passes the **id** as an argument in every call of the function **CommitPerformanceRecord**, along with the following parameters: a) the required *activity*, b) its *scope* (i.e., specific or general), c) *score* representing the child's play performance, d) relevant *comments*, and e) captured *footage.* At the end of the play, the smart game calls the **EndSession** passing the *sessionId* as parameter in order to notify that the current active session has ended. Further functionality is available to smart games. More specifically:

- **CalculateUserCapacityInActivity:** calculates and returns the estimated capacity for the execution of a child's required activity. This process uses every play performance score, regardless of its scope (i.e., specific or general) and the smart game which has originated it.

- **CalculateUserCapacityInActivityInGame:** offers the same functionality as the previous one, but it uses only scores originated from a certain smart game which is given as parameter.
- **CalculateUserCapacityInActivityWithScopeInGame:** offers the same functionality as *CalculateUserCapacityInActivity,* but it uses only scores originated from a certain smart game which is given as parameter and they have a selected scope (i.e., specific or general) given as an argument.
- **HasUserLimitationInActivity:** returns *TRUE* if the child has limitation in execution of an activity given as parameter according to the child's profile (e.g., *orientation in space*).
- **HasUserLimitationOfBodyFunction:** returns *TRUE* if the child has functional problem regarding a body function given as parameter (e.g., *hearing*).
- **IsUserCapable***:* takes as arguments a list of required activities and a list of body functions and returns TRUE if the function *HasUserLimitationOfBodyFunction* returns false for every body function given as parameter AND (&&) the function *HasUserLimitationInActivity* returns false for every given activity*.*
- **HasUserSufficientCapacityInActivity:** takes the following parameters: a) the required activity, and b) a threshold for the required capacity. It returns *TRUE* if the required activity's estimated capacity is higher than the argument. The process calls the aforementioned functions in order to verify that the child has neither limitation nor body structure functional problems regarding the given activity and the involved body functions. In case of lack of existing interaction history, the process returns *TRUE* if the expected capacity of a specific child's age (according to the Denver II scale) is above the capacity threshold given as a parameter.
- **ComputeInferredUserAgeInGame**: computes and returns the inferred child's age of maturation ignoring his actual one by taking into account the interaction history (i.e., play performance scores) originated from the smart game given as a parameter. In case of lack of existing interaction history the process returns the child's actual age. In detail, the process analyses the play performance scores regarding the specific required activity that is targeted of a certain level of the game and extracts its capacity based on the aforementioned forecasting methods. Supposing that the capacity's predicted value is greater than the previous one and above the threshold of 90%, the process repeats the assessment for the specific required activity of the next level of the smart game. Similarly, the process repeats the assessment for the specific required activity of the previous level if the capacity's predicted value is lower than the previous one and below the threshold of 20%.
- **GetAssociatedSystemTasksForActivity**: returns a list of system tasks that have been associated with the activity given as parameter.
- **GetAssociatedHumanTasksForActivity**: returns a list of human tasks that have been associated with the activity given as parameter.

Regarding the employed adaptation suggestions, ADAM monitors the inferred child's age of maturity and notifies the selected smart game for any change. In detail, it computes the inferred child's age every time a play performance score, of a specific required activity, is committed by a smart game. ADAM monitors whether the inferred age has been changed after the commitment and subsequently sends a notification to the smart game through the function **Event_InferredUserAgeInGameChanged**. Similarly, it repeats the same process for each required activity whose capacity has changed significantly since the commitment of the corresponding performance play score through the function **Event_InferredActivityCapacityInGameChanged**.

More types of adaptation suggestions are being introduced by ADAM in order to facilitate automatic interaction adaptation (i.e., the efficient selection of available modalities) based on children's skills and abilities that continuously change. For instance, ADAM monitors the received capacity of a selected activity (e.g., *watching*) during play and in case of immaturity indication it suggests smart game to try different modalities (i.e. output communication channels) through the function **Event_TryDifferentMultimodality** (e.g. presentation of still images can be switched to the presentation of video). In detail, ADAM suggests smart game one of the activity's associated system tasks (according to the **Bean model**) and then continues to monitor activity's capacity variations. In case of the immaturity indication remains, it repeats the same process until the trial of all the associated system tasks for the selected activity. When deemed necessary (i.e., all the associated system tasks for the selected activity have been tried), ADAM suggests the alteration of the assessed activity with an equivalent one from the same category (according to the **knowledge base**) through the function **Event_TryAssessEquivalentActivity**. For instance, when all modalities (i.e., system tasks) for the activity *watching* have been tested, ADAM suggests *listening* as a new activity for assessment which origins from the same activity category (i.e., *Purposeful sensory experiences*) as *watching* does.

4.4 Nibbler: A sensory infrastructure to support interaction and interaction monitoring

This section presents the design and implementation of a unified sensory infrastructure mechanism, called Nibbler, which supports a number of alternative natural interaction techniques through the integration of facilities to support: a) gestures recording, parameterization and recognition, b) face tracking, c) head position estimation, d) skeleton tracking and e) speech recognition. To this end, Nibbler facilitates the developed framework to provide supplementary forms of interaction for children, thus enhancing the play experience. For example, smart games are provided with biometric information, rich interaction metaphors and speech input.

Nibbler builds on the Microsoft Kinect sensor and was developed using the C# programming language, Microsoft Kinect software development kit (SDK) v1.8 and the .NET Framework. The GUI of the proposed software platform is presented in Figure 41.

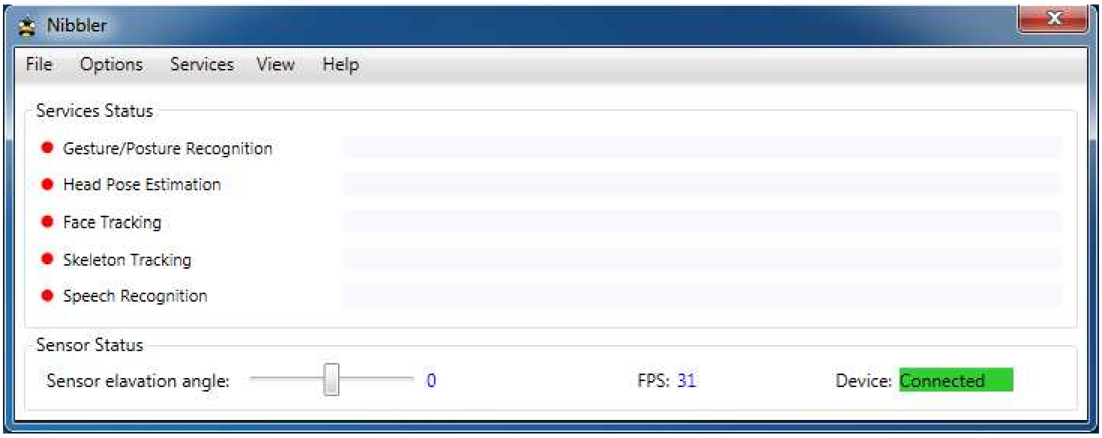


Figure 41: Nibbler; an infrastructure implementing the most of sensory services

4.4.1 Sensory modules

Nibbler is organized into various modules, each of which is responsible for specific sensory requirements. Each module is presented in detail in the following sections.

4.4.1.1 Skeleton tracking module

The **skeleton tracking** module is responsible for reporting position information of each skeleton joint. This module performs geometric transformations on each skeleton joint position constituting every real time skeleton frame. This happens in order to get the same valid results regardless of the position of the user who may be located everywhere inside the sensor's field of view.

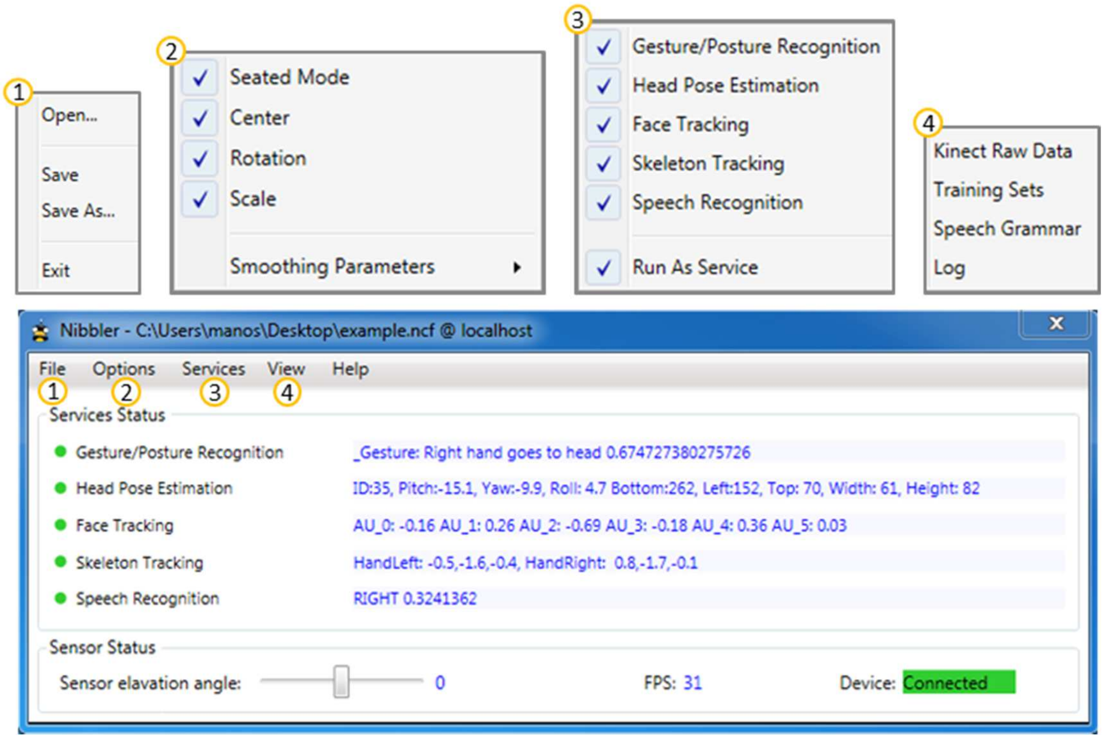


Figure 42: Nibbler in a fully working mode

The skeleton tracking module transforms the user's skeleton to a local scope, i.e., expressed relatively to the 3-axis coordination system centered in the middle of the user's shoulders. The transformation is applied with respect to three distinct steps, translation, scaling and rotation, as shown in Figure 47②. Firstly, each joint's x-axis position is subtracted with the position dynamically calculated as the center of both shoulders. This way, skeleton tracking is performed regardless of the user's relative position to the sensor, as presented in Figure 43. Secondly, the joints' positions are normalized in order to be scale-independent. Finally, the module rotates the skeleton so as to align the user's skeleton to the sensor. This is accomplished by multiplying each joint's position with a matrix calculated from the angle θ, where θ is equal to –yaw (yaw is the angle of line between the right and the left shoulder).

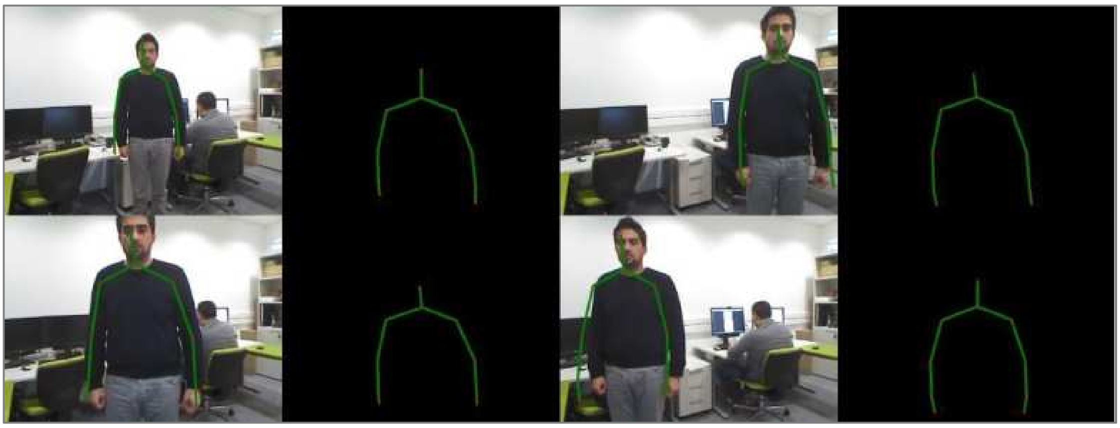


Figure 43 : An example of position independence between user and sensor.

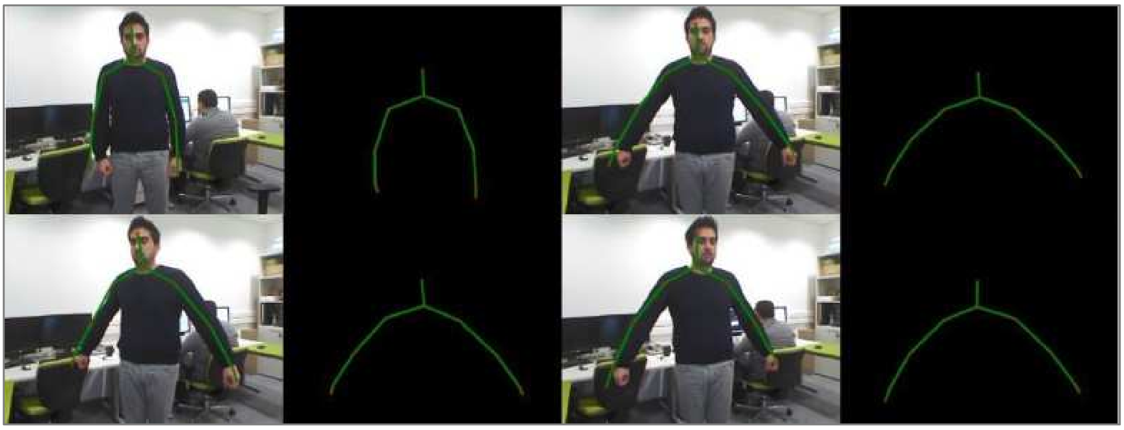


Figure 44: An example of alignment independence between user and sensor.

4.4.1.2 Gesture/Posture recognition module

Although much work has been done in the domain of Gesture/Posture recognition, there are no ready to use solutions widely available to developers aiming to incorporate Kinect based gesture recognition in their applications. However, the Microsoft Kinect SDK provides a concrete example containing sample source code for the next and previous gestures. To this end, this part of the module partially replicates existing works in gestures recognition building on well-established practices in the domain so as to integrate this very important form of natural interaction to the developed framework. The Gesture/Posture recognition module implements the dynamic time warping (DTW) algorithm for measuring similarity between two skeleton sequences which may vary in time or speed. The most important contributions of this module is the provision of a training platform that allows developers to fine tune their gestures by having access to a number of alternative biometric parameters. In general, the DTW algorithm can be applied to any data which can be turned into a linear sequence. A well-known application has been automatic speech recognition, to cope with different speaking speeds[44]. The **first skeleton sequence** is captured and fine-tuned only once during the training process, while the **second** one is captured constantly in real time. During the training process, the author (e.g., end users from UG4: Game Developers) is able to record a skeleton sequence using his own body as input data and store it in the database (see Figure 45). The number of the frames in a sequence may vary from 1 up to a maximum predefined variable, which is usually 30 considering that 30 is the maximum sensor's frame rate according to its specification details.

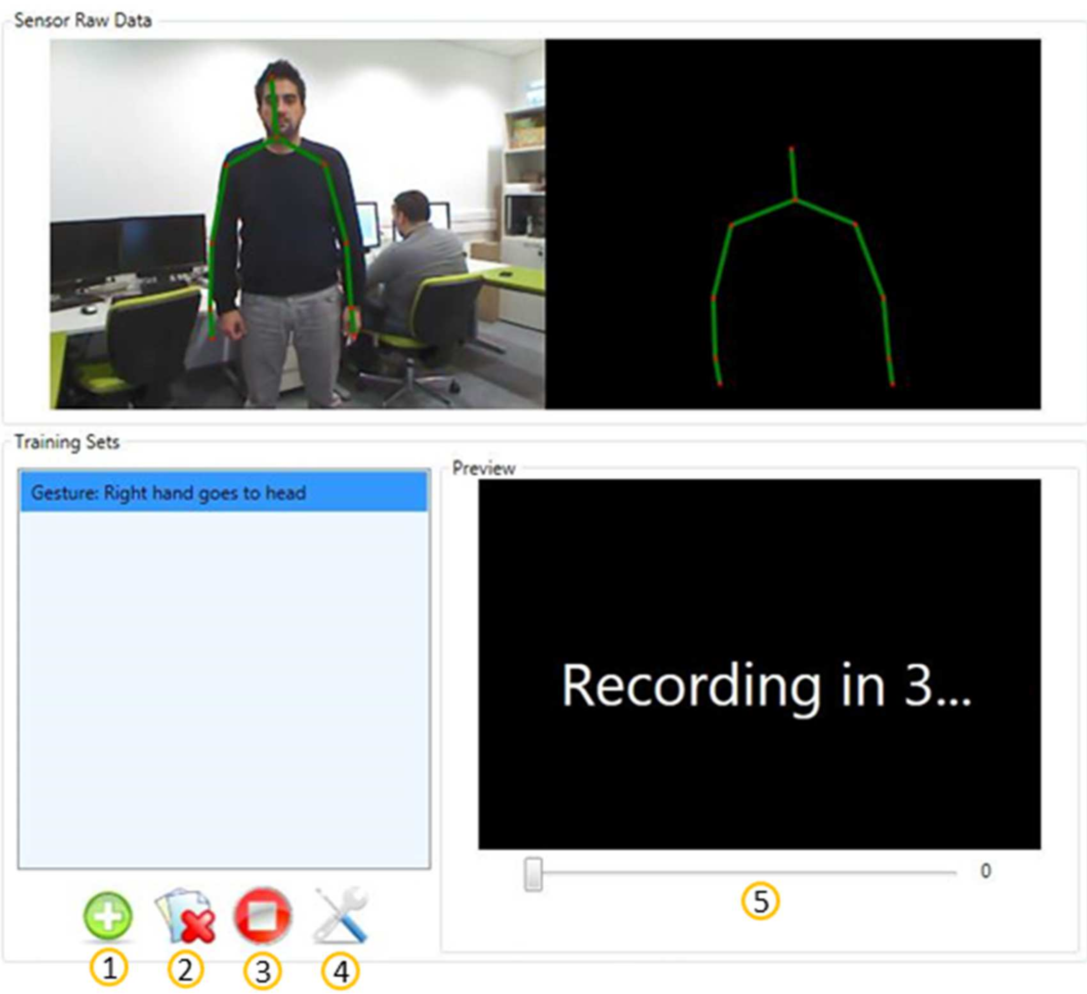


Figure 45: The author starts recording a skeleton sequence in seated mode (i.e., only the upper half skeleton is captured)

[44] http://en.wikipedia.org/wiki/Dynamic_time_warping

End users are provided with a plethora of functions to optimally adjust gesture/posture recognition. In particular, end users may (see Figure 45) add a new gesture using button①, delete an existing one using button②, or start/stop skeleton sequence recording using ③. When recording is completed, the author is able to preview the recorded skeleton sequence, edit it and fine tune the captured skeleton sequence by pressing button ④ and using the pop up configuration window as shown in Figure 47. This window offers functionality for: a) renaming ①, b) adjusting the maximum distance with the real time skeleton sequence ② for successful recognition (see below for further details), c) modifying the number of the minimum frames ③ that have to be captured in real time before the recognition process is triggered, d) trimming the corresponding skeleton sequence ⑤, ⑥ and e) adjusting some of the basic parameters used in the DTW algorithm such as the slop constraint which determines the maximum slope in the optimal path ⑦. Additionally, the author can select only the joints which mainly characterize a gesture ⑧, i.e., when the goal is to recognize a gesture in which the user uses his right hand to select the next photo by slightly moving it from right to left, the remaining joints of the body do not need to be taken into account. Furthermore, if some axis doesn't play an important role for a gesture, such as the Z axis (the axis of depth) in the aforementioned example, the author can disable it by unselecting the corresponding checkbox④. Lastly, a gesture playback panel ⑨ is available to allow the author preview the recorded skeleton sequence in front and side realization.

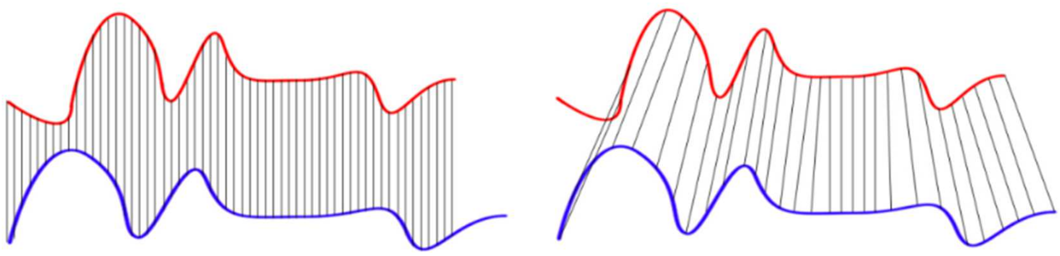


Figure 46: Euclidean vs. Dynamic Time Warping Matching

When the recognition module is running, it captures constantly, in real time, skeleton frames, and when their total number reaches the number equivalent to one second, it then starts the matching process. The latter calculates an optimal distance between the real time sequence and every sequence which is stored in the database. The sequences are "warped" non-linearly in the time dimension to determine a measure of their similarity independent of certain non-linear variations in the time dimension. Dynamic Time Warping (DTW) allows elastic shifting in the time domain and matches sequences that are similar but out of phase as shown in Figure 46.

4.4.1.3 Head pose estimation/face tracking module

The head pose estimation/face tracking module enables the creation of smart games based on natural and intuitive interaction styles. The implementation of this module builds on the Microsoft Kinect SDK and the provided facilities to track human faces in real time. The module reports measurements about the three axes of rotation of user's head pose (pitch, roll, and yaw) based on a right-handed coordinate system. Additionally, it reports the rectangle enclosing the head in the captured frame. For face tracking, six animation units (AUs) are tracked in real-time which are a subset of what is defined in the Candide3 model[45]. The results are expressed in terms of numeric weights which are deltas from the neutral shape varying between -1 and +1.

4.4.1.4 Speech recognition module

The speech recognition module is based on the Microsoft.Speech library found in the Microsoft Kinect SDK and the sensor's built-in microphone array as voice input device. This module takes a grammar as input and loads it in the recognition engine. Currently, speech recognition is supported only in few languages. The module reports information about the spoken word or phrase which contains a unique tag, confidence as well as the speaking duration.

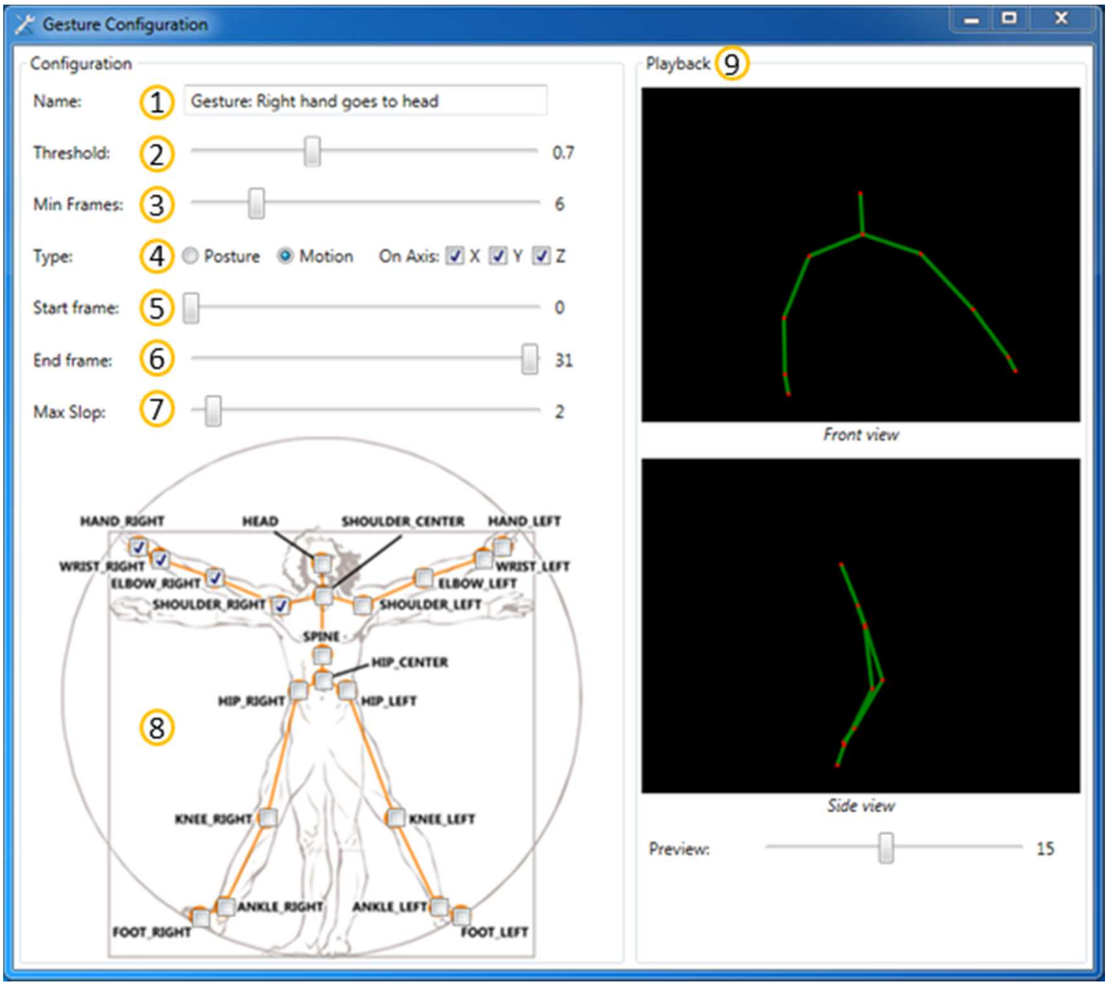


Figure 47: Configuration and fine tuning of a gesture

4.4.2 Using Nibbler from a developer's perspective

The presented modules integrated in the Nibbler sensory infrastructure run simultaneously without any performance issues at almost 30fps on an ordinary pc (see Figure 42). Figure 42 illustrates the settings that are available for configuring Nibbler for the desired context of use. Nibbler communicates with clients (e.g., smart games)

[45] http://www.icg.isy.liu.se/candide/

and reports each module's measurements via a middleware network layer. The latter is designed to facilitate the communication of systems that are deployed on diverse platforms as presented in [76]. Additionally, Nibbler can accept requests from clients in real time to change either the gesture training set or the grammar used for speech recognition. In this context, Nibbler's functionality is described in the interface definition language (IDL[46]) as follows:

Definitions

```
enum JointTrackingState { Inferred, NotTracked, Tracked};

enum JointType {
              HipCenter,
              Spine,
              ShoulderCenter,
              Head,
              ShoulderLeft,
              ElbowLeft,
              WristLeft,
              HandLeft,
              ShoulderRight,
              ElbowRight,
              WristRight,
              HandRight,
              HipLeft,
              KneeLeft,
              AnkleLeft,
              FootLeft,
              HipRight,
              KneeRight,
              AnkleRight,
              FootRight
};

struct Point3D
{
       double X;
       double Y;
       double Z;
};

struct Joint
{
       JointTrackingState trackingState;
       JointType type;
       Point3D position;
};

struct SemanticResultValue
{
       string phrase;
       string value;
};

typedef sequence<SemanticResultValue> SemanticResultValueSeq;
```

Gesture recogniti on

```
ami::StringSeq GetGestureNames ();
```

```
boolean LoadGestures(in ami::OctetSeq gesturesConfigStream);
```

```
void Event_GestureRecognized (in string gesture, in double
distance);
```

[46] http://en.wikipedia.org/wiki/Interface_description_language

	`void Event_GestureRecognizedExt (in string gesture, in double distance, in ami::OctetSeq colorImgStream);`
Head pose estimation	`void Event_HeadPoseChanged (in double pitch, in double yaw, in double roll);`
	`void Event_HeadRectChanged (in long left, in long bottom, in long top, in long width, in long height);`
Face tracking	`void Event_FaceAnimationUnitsChanged (in ami::FloatSeq faceAnimationUnits);`
Skeleton tracking	`void Event_HandRightPositionChanged (in Point3D position);`
	`void Event_HandLeftPositionChanged (in Point3D position);`
	`void Event_SkeletonChanged (in JointSeq joints);`
	`void Event_SkeletonChanged (in JointSeq joints);`
Speech Recognition	`boolean LoadGrammar(in SemanticResultValueSeq grammar, in string culture);`
	`void Event_SpeechRecognized (in string value, in float confidence, in long long duration);`
	`void Event_SpeechRejected ();`

CodeBlock 2: Nibbler's service oriented functionality (IDL)

4.5 A collection of augmented artifacts

In the context of AmI, interactive devices must be unobtrusive, hidden or embedded in traditional everyday objects and furniture augmented with ICT technology without compromising general health and safety requirements. To support the goals of the current thesis, an artifact-oriented approach has been adopted. As presented in 3.2.1, a collection of augmented artifacts was designed and implemented aiming to be deployed in an environment surrounding young children (e.g., a child room or a kindergarten).

4.5.1 Beantable: An augmented interactive table

Beantable is an augmented interactive table, whose original design has been presented in section 3.2.11 (see Figure 29 and Figure 30). The physical construction of the augmented interactive table was completed in close collaboration with a carpenter (see Figure 49). Beantable is appropriate for use by children aged from 3 to 6 years old. The table is a wooden prototype with dimensions of 116cm (L) X 105cm (W) X 46cm (H), and has been implemented to be robust yet transferrable. It was designed and constructed to afford to be used as any standard table for eating, drinking, and other common activities.

Figure 48: Beantable: The augmented interactive table

Using back projection, a main display device with a standard resolution of 1024 X 786 pixels is located on the top side of the table surface and has dimensions of 56cm (L) X 40cm (W). Choosing the appropriate translucent surface was a challenging process. The most suitable surface should provide a crispy image, while at the same time allowing a clear image of the physical objects placed on it, in order to be captured by the cameras. Additionally, the surface had to be as robust as possible. To this end, various types of surfaces of different thicknesses (3mm, 5mm, and 1cm) were tested including plexiglass, sandblasted glass, and normal glass covered with window frosted glass film (see Figure 50). Taking young children's safety into account, the selected surface was a 6mm high robust triplex (laminated) glass. This type of glass is created by bonding two or more glass panes using a polymer membrane – PVB (see Figure 50). The bonding of the glass is realized by heating and subsequently applying pressure. The triplex glass exhibits maximum resistance to shocks and stresses. This glazing system is distinguished for providing high safety, because when broken the pieces of glass are held in place by the intermediate membranes, preventing the fall of sharp pieces of glass, which would potentially cause serious injuries. To support back projection, the glass was covered with a window frosted glass film.

Figure 49: The interactive table during construction

The main display device is enabled with multi-touch and force-pressure sensitive capabilities, and is able to recognize the location and rotation of physical objects that are placed on it provided that each of them carries at least one fiducial marker at its bottom side. Fiducial markers are visual patterns printed on paper which have topological characteristics that make them to detect and track through visual recognition algorithms. The minimum size of each fiducial marker is 3x3 cm.

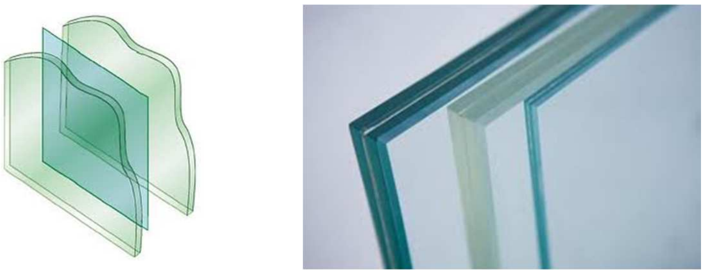


Figure 50: High robust triplex glass was selected for the main display device

Beantable is complemented by a custom made chair, with dimensions of 68cm (L) X 51cm (W) X 78cm (H), which is able to detect when the child is sitting on it and capture body posture data. Both the table and the chair have a soft yellow color which according to pedagogical experts is neutral and easily recognizable to children (see Figure 48). Additionally, the surface is able to capture information about the pressure exerted by the user during interactions by installing 4 force-pressure sensors beneath it as presented in section 4.5.1.4.

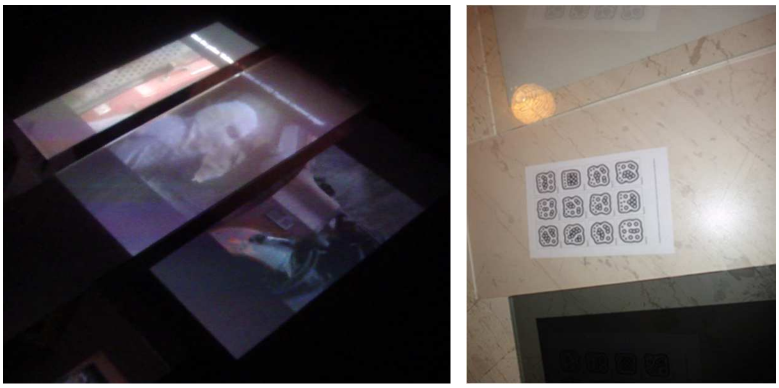


Figure 51: Testing various types of surfaces

4.5.1.1 Hardware infrastructure

Regarding the hardware infrastructure, all the necessary components are embedded and not visible to end users. As previously mentioned, Beantable provides the AmI environment with the following AmI services: a) multi-touch through fingertip recognition, b) physical object recognition, c) force pressure recognition onto table's surface and d) force pressure recognition while seating. Figure 52 and Figure 53 show all the hardware components installed inside the artifact for the operation of the aforementioned services. In detail, the embedded hardware infrastructure includes:

- A high performance desktop computer equipped with high end graphics card. For the current prototype a Dell OptiPlex 9020 Mini Tower[47] was selected with the following characteristics: a) a 4th Generation Intel® Core™ i7 Processor (Quad Core, 8MB, 3.60GHz), b) Windows 7 Professional English, c) 8GB (2x4GB) 1600MHz DDR3 Memory, d) 1TB 3.5inch SATA (7.200 RPM) Hard Drive, and e) NVIDIA GeForce GT 645.
- LG HX300G[48] which is a micro portable projector. It weighs only 1.7 pounds and it measures only 2.4" x 6.3" x 5.3", small enough to embed inside Beantable. It produces 300 lumens of brightness in its brightest mode and has 2000:1 contrast. The native resolution is XGA.
- Two Flea2G 1.3 MP Mono FireWire 1394b (Sony ICX445)[49] high quality cameras located behind the main display device equipped with wide lens to maximize the quality of the captured image. Each camera has a standard 9-pin IEEE-1394b connector that is used for data transmission, camera control and powering the camera. The resolution of the captured image is 1288x964 (in pixels) at 30 FPS.
- Four infrared illuminators (or IR illuminators) located behind the main display device covered with thin fabric or mat film to diffuse the light smoothly. Each illuminator has 48 infrared led of a life time over 10000 hrs while it operates at 12V DC.
- Two cooling fans located at specific points that facilitate the natural air flow. Each fan offers extreme high airflow and static pressure. It operates at DC 12V while its maximum speed is 5500 rpm. Other features are: a) 4 wire with PWM Control, b) Speed sensor, and c) dimensions of 120mm x 120mm x 38mm.
- Two DS18B20[50] temperature sensors. DS18B20 is a one-wire digital temperature sensor that provides 9-bit to 12-bit Celsius temperature measurements. The DS18B20 communicates over a 1-Wire bus that by definition requires only one data line (and ground) for communication with a central microprocessor. It has an operating temperature range of -55°C to +125°C and is accurate to ±0.5°C over the range of -10°C to +85°C.
- Two speakers taken from a Dell AS500 sound bar that fits to some Dell flat panel LCD monitors. Each speaker features output of 5 Watts. The sound bar was

[47] http://www.dell.com/us/business/p/optiplex-9020-desktop/pd
[48] http://www.projectorcentral.com/LG-HX300G.htm
[49] http://www.ptgrey.com/flea2g-13-mp-mono-firewire-1394b-sony-icx445-camera
[50] http://datasheets.maximintegrated.com/en/ds/DS18B20.pdf

disassembled in order to place each speaker separately as depicted in Figure 53. The sound bar's embedded controller was also installed next to the desktop computer as seen in the same figure and operates at 12V DC.

- Four FlexiForce[51] pressure sensors located just behind the screen. The FlexiForce sensor can measure force between almost any two surfaces and is durable enough to stand up to most environments. The installed sensors measure up to 25 lb of force. Further description is presented in 4.5.1.4.

Figure 52: The artifact with the left cover opened

Figure 53: The artifact with the right cover opened

[51] http://www.phidgets.com/products.php?product_id=3102

4.5.1.2 Embedded cooling system

As presented in 4.5.1.1, two cooling fans are embedded inside Beantable in order to facilitate the natural air flow. The fans aim to cool embedded hardware components by moving heated air away from them and draw cooler air over them. The fans attached to Beantable are used in combination with the installed temperature sensors in order to improve the efficiency of cooling. A microcontroller reads current temperature sensors value and adjusts accordingly the speed of the installed fans. As a result, they shut down when Beantable is turned off or the temperature is below 25°C. The selected microcontroller is an Arduino Uno[52] and uses RS232 as an interface to the embedded desktop computer communication.

A window's service runs continually on the Beantable's embedded computer in order to communicate with the microcontroller and exposes control functionality to the AmI environment. Thus, administrators can easily configure the microcontroller and be notified in case of hardware malfunction or overheating. The service communicates with AmI clients and reports temperature measurements and fan speed information via a middleware network layer. Additionally, it can accept requests from clients in real time to change either the speed of a fan or set the temperature threshold for overheating alarm. Such functionality is described in the interface definition language (IDL) as follows:

```
long GetCountOfTemperatureSensors();
float GetTemperatureForSensorWithIndex(in long sensorIndex);

long GetCountOfCoolingFans();
long GetSpeedOfFanWithIndex(in long fanIndex);

//speed value 0-100
void SetSpeedToFanWithIndex(in long fanIndex, in long speed);
void SetSpeedToAllFans(in long speed);

void EnableAutoCooling(in long temperature);
void DisableAutoCooling();
long GetAutoCoolingStatus();

void TurnOffCoolingSystem();
void TurnOnCoolingSystem();

void SetAlertOn(in long temperature);
void SetAlertOff();
long GetAlert();

//
// Events
//
void Event_OnHeatAlert(in long averageTemp);
void Event_OnAverageTempChanged(in float averageTemp);
void Event_OnFanSpeedChanged(in long fanIndex, in long speed);
```

CodeBlock 3: Cooling system implemented functionality (IDL)

The service communicates via RS232 with the microcontrollers and provides the AmI clients with the following functionality:

[52] http://arduino.cc/en/Main/arduinoBoardUno

- **GetCountOfTemperatureSensors**: returns the count of the installed temperature sensors.
- **GetTemperatureForSensorWithIndex**: returns the measurement of a selected temperature sensor given as a parameter.
- **GetCountOfCoolingFans**: returns the count of the installed cooling fans.
- **GetSpeedOfFanWithIndex**: returns the speed in rpm of a selected fan given as a parameter.
- **SetSpeedToFanWithIndex:** sets the speed of a selected fan to a value in rpm given as parameter.
- **SetSpeedToAllFans:** sets the speed of all the installed fans to a value in rpm given as parameter.
- **EnableAutoCooling:** allows the microcontroller to automatically adjust fans' speed according to the operating temperature. Higher temperature has as a result cooling fans to operate at a higher speed.
- **DisableAutoCooling:** does not allow the microcontroller to automatically adjust fans' speed according to the operating temperature.
- **GetAutoCoolingStatus:** returns *TRUE* if the microcontroller is configured to automatically adjust fans' speed according to the operating temperature
- **TurnOffCoolingSystem:** deactivates cooling fans (i.e. used in case of Beantable turns off).
- **TurnOnCoolingSystem:** activates cooling fans (i.e. when Beantable turns on).
- **SetAlertOn:** enables monitoring of the Beantable's operating temperature and in case of its value is above the threshold (given as a parameter) a notification will be sent to the AmI clients via the **Event_OnHeatAlert.**
- **SetAlertOff:** disables monitoring of the Beantable's operating temperature.
- **GetAlert:** returns the threshold for alert overheating.
- **Event_OnAverageTempChanged:** notifies AmI clients in any change of the operating temperature (calculates the average of all the installed temperature sensors).
- **Event_OnFanSpeedChanged:** notifies AmI clients in case of a fan's operating speed changed significantly.

4.5.1.3 Physical object and cursor recognition

Beantable consists of a translucent display device which allows the embedded cameras to capture clear images of the physical objects placed on it. Physical object recognition is facilitated by the use of appropriate computer vision sub-systems. As depicted in Figure 54, these computer vision sub-systems are: a) **CvUtis**, b) **VirtualCamera**, c) **ImageNormalizer**, and d) **reacTIVision**[53]. The latter is an open source, cross-platform computer vision framework for the fast and robust tracking of fiducial markers attached onto physical objects, as well as for multi-touch finger tracking. **reacTIVision** sends TUIO messages via UDP in order to transmit physical object recognition information to the **"Physical Object and Cursor Recognition"** TUIO client. The same protocol is used

[53] http://reactivision.sourceforge.net/

to translate touch actions identified on top of the table surface to native touch events forwarded by the employed **MultiTouchVista**[54] (UniSoftHID) driver. The "**Physical Object and Cursor Recognition"** service is built upon a middleware network layer in the interest of making available object and cursor recognition information to AmI clients. The aforementioned computer vision sub-components along with some utilities used for calibration and configuration purposes are discussed below.

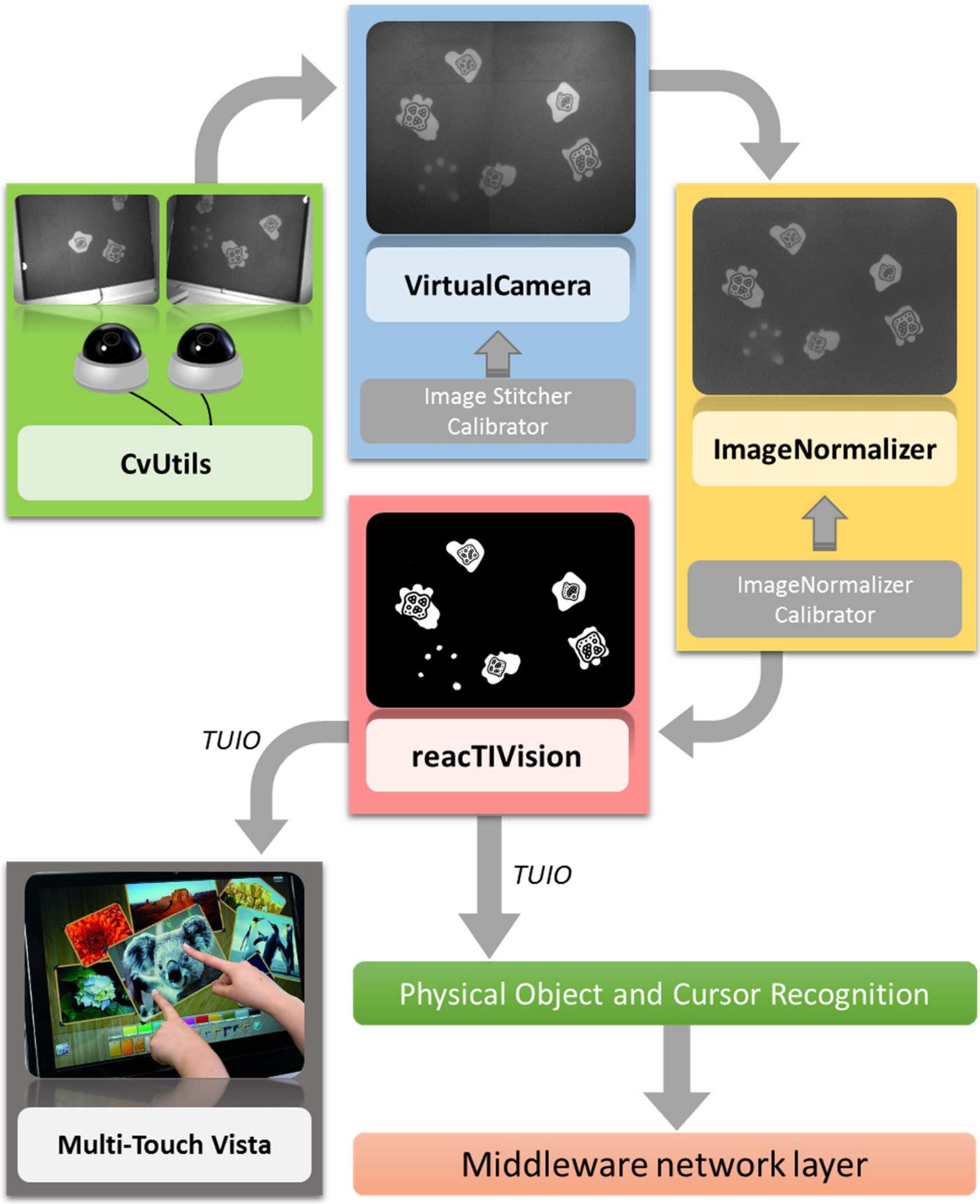


Figure 54: Computer vision sub-systems used for physical object and cursor recognition.

[54] https://multitouchvista.codeplex.com/

4.5.1.3.1 Computer vision sub-systems

The **CvUtils** computer vision sub-system is responsible to get captured images from each connected camera (i.e., Flea2G 1.3 MP Mono FireWire 1394b). **CvUtils** processes each captured frame (with resolution of 1280x960 pixels) in order to produce its undistorted version (see Figure 55). Subsequently, it servers each processed frame to other applications by using a shared memory address. The latter is mapped to a specific context name (e.g., "*PGRCam*") and is used to temporarily store image data which can be easily read by other processes.

The **VirtualCamera** is a computer vision software responsible for image stitching and generation of a panoramic ortho-canonical image with respect to a plane. The software uses input images captured from calibrated monochrome or color cameras. It reads the images from the shared memory where they written by **CvUtils** (e.g., "*PGRCam*"). After processing, it writes to shared memory the ortho-canonical panoramic image (with resolution of 1672x1254 pixels) to be used by other programs (e.g., "*VirCam*"). To produce optimal results the software requires camera calibration and parameters file generation before use. **Image Stitcher Calibrator** is a computer vision software responsible for the computation of a parametric file for image stitching and generation of a panoramic ortho-canonical image with respect to a plane. The software uses input images captured from calibrated monochrome or color cameras. It reads the images from shared memory. The user selects for each camera rectangular areas to stitch on an image, using the aid of an object of known size (e.g., a printed checkerboard). Such process produces the parametric file "ImageStitcher.xml" to be used by the **VirtualCamera** software.

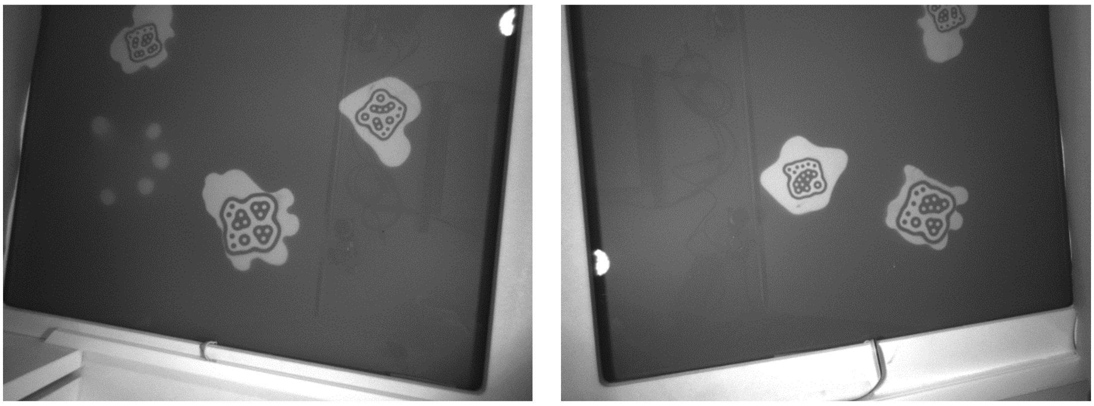


Figure 55 : CvUtils captured images (two cameras with wide lens are used)

The next computer vision sub-system **ImageNormalizer** reads the images from the **VirtualCamera** shared memory (e.g., "*VirCam*") and proceeds to image normalization (see Figure 56). During such process, **ImageNormalizer** changes the range of pixel intensity values in order to equalize poor contrast due to glare. This process is necessary due to the non-uniform illumination produced by the embedded infrared beacons. Additionally, it applies Gaussian filter[55] (i.e., a diverse linear filter) for smoothing the stitched image in order to reduce noise. Gaussian filtering is done by convolving each point in the input array with a Gaussian kernel and then summing them all to produce

[55] http://en.wikipedia.org/wiki/Gaussian_filter

the output array. After processing, **ImageNormalizer** writes the normalized image to shared memory in order to be used by other programs (e.g., "*NorCam*").

ImageNormalizer Calibrator is the utility calibration used in **ImageNormalizer** and produces a configuration file with intensity transformations of every pixel. In detail, the Beantable's surface is covered by a calibration board. The **ImageNormalizer Calibrator** reads the intensity of every pixel and estimates the average intensity (e.g., 140 in the range 0-255). Then, it calculates and writes to a configuration file a value which implies to the difference between the intensity average and the intensity value of every pixel. **ImageNormalizer** uses that configuration file to process every input image frame.

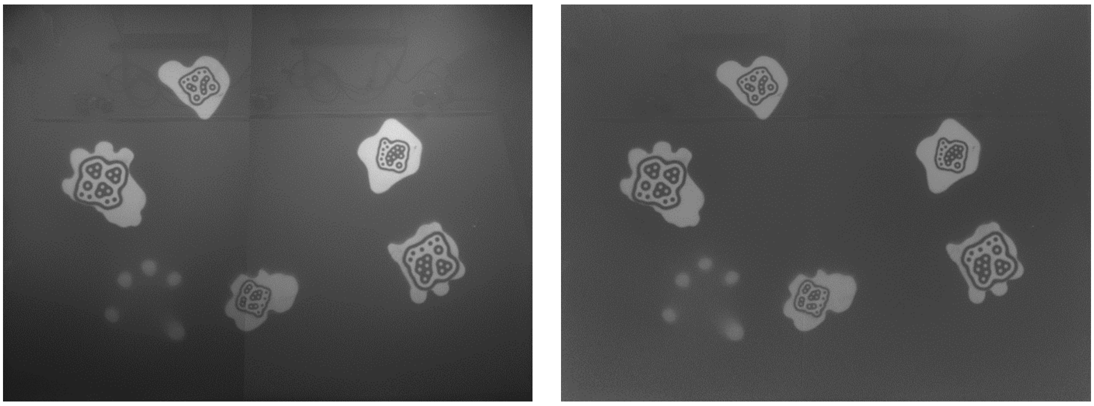


Figure 56 : Stiched image as output from VirtualCamera (Left). Normalized image as output from ImageNormalizer (Right).

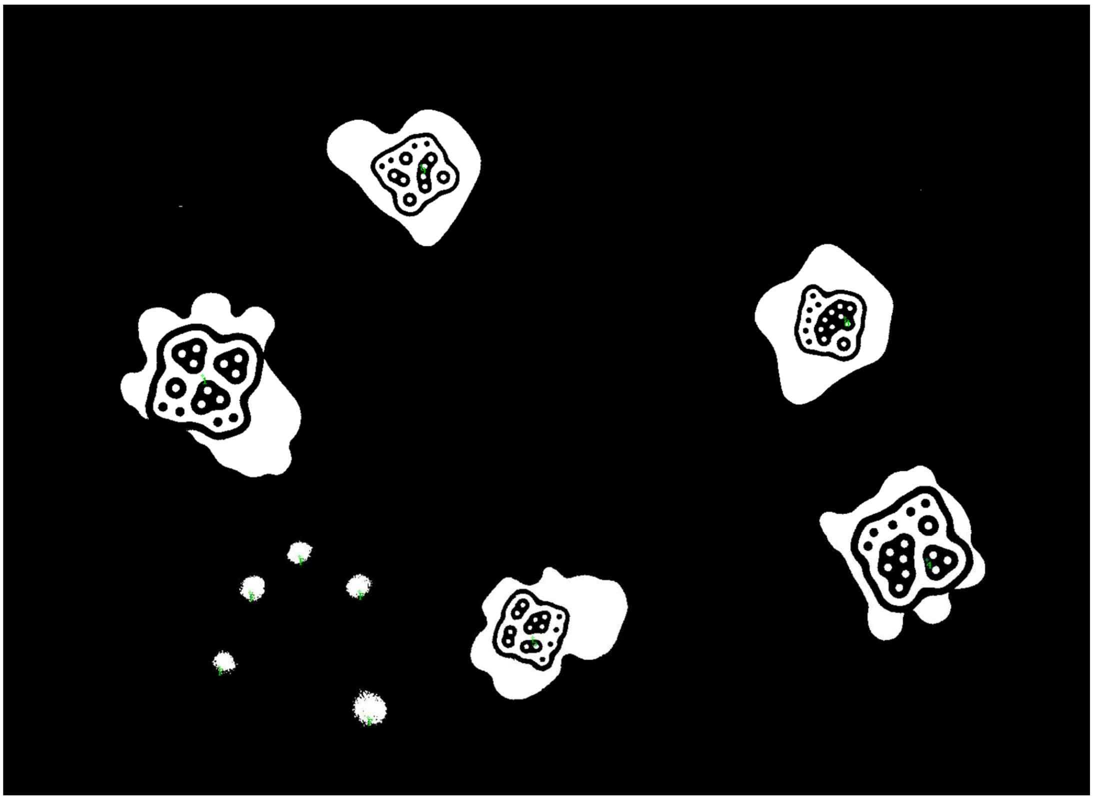


Figure 57: reacTIVision reads normalized images from shared memory and proceeds with physical object and cursor recognition.

Finally, a modified version of the open source, cross-platform computer vision framework **reacTIVision** is employed for physical object and cursor recognition. **reacTIVision** is used for the fast and robust tracking of fiducial markers attached to physical objects, as well as for multi-touch finger tracking (see Figure 57). For the purposes of the current work, the version 1.4 (published on 19-MAY-2009) was modified in order to enable image reading from a shared memory. In depth, two functionalities were added: a) a new type of camera, named "SharedMemoryCamera", and b) an image binarization/segmentation technique. To this end, **reacTIVision** reads images written by the **ImageNormalizer,** (e.g. "*NorCam*") from the shared memory and processes them in order to produce a binarized image in which each pixel is stored as a single bit (i.e., 0 or 255). At the same time, **reacTIVision's** internal gradient functionality is disabled, and the process continues with the binarized image in order to retrieve and identify fiducial markers or small round white blobs as finger tips on the surface.

The **reacTIVision** application sends the TUIO messages to both the Multi-Touch Vista driver and the "**Physical Object and Cursor Recognition"** service. Multi-Touch Vista is a user input management layer that handles input from various devices (touchlib, multiple mice, TUIO, etc.) and normalizes it against the scale and rotation of the target window. The "**Physical Object and Cursor Recognition"** service is described in the next section.

4.5.1.3.2 Physical object and cursor recognition service from a developer's perspective

Physical Object and Cursor Recognition service makes available to AmI clients information about physical object and cursor recognition. In detail, it uses the TUIO protocol to receive information from the aforementioned computer vision sub-system **reacTIVision**. The service is implemented in C++ and uses the TUIO v1.1 C++ library[56] and communicates with AmI clients via a middleware network layer. The delivered functionality is described in the interface definition language (IDL) as follows:

```
//
// Types
//
enum CursorState { C_ADDED, C_ACCELERATING , C_DECELERATING, C_STOPPED,
C_REMOVED};
enum PhysicalObjectState { FO_ADDED, FO_ACCELERATING , FO_DECELERATING,
FO_STOPPED, FO_REMOVED, FO_ROTATING};

struct Time
{
	long long seconds;
	long long micro_seconds;
};

struct Point
{
	float xpos;
	float ypos;
	Time currentTime;
```

[56] http://www.tuio.org/?cpp

```
        Time startTime;
};

typedef sequence<Point> PointSeq;

struct Container
{
        long long session_id;
        float x_speed;
        float y_speed;
        float motion_speed;
        float motion_accel;
};

struct Cursor
{
        long cursor_id;
        Point position;
        Container info;
        CursorState state;
};

struct PhysicalObject
{
        long symbol_id;
        float angle;
        float rotation_speed;
        float rotation_accel;
        Point position;
        Container info;
        PhysicalObjectState state;
};

typedef sequence<Cursor> CursorSeq;
typedef sequence<PhysicalObject> PhysicalObjectSeq;
//
// Methods
//
CursorSeq getCursors();
PhysicalObjectSeq getPhysicalObjects();

//
// Events
//
void Event_PhysicalObjectAdded(in PhysicalObject obj);
void Event_PhysicalObjectUpdated(in PhysicalObject obj);
void Event_PhysicalObjectRemoved(in PhysicalObject obj);
void Event_CursorAdded(in Cursor cur);
void Event_CursorUpdated(in Cursor cur);
void Event_CursorRemoved(in Cursor cur);
```

CodeBlock 4: Physical object and cursor recognition service. Implemented functionality (IDL)

Regarding cursor recognition, the **Physical Object and Cursor Recognition** service receives UDP data of the TUIO protocol specification about cursors/touches happened onto the Beantable's surface. Cursor information consists of: a) cursor tracking id, b) position of the cursor (two floating point values in a range of 0-1 representing a percentage scale within image's size), c) cursor's state (i.e., C_ADDED, C_ACCELERATING , C_DECELERATING, C_STOPPED, C_REMOVED), and d) information about motion acceleration. The service offers to AmI clients the aforementioned cursor information by exposing the following functionality:

- **getCursors**: returns to the AmI client a list (i.e. "**CursorSeq**") of currently active cursors.
- **Event_CursorAdded:** notifies AmI clients when a new cursor is added. In other words, when user touches the Beantable's surface.
- **Event_CursorUpdated:** notifies AmI clients when an existing cursor is updated. For example, when a user moves his finger along the Beantable's surface.
- **Event_CursorRemoved:** notifies AmI clients when an existing cursor is removed. For example, when a user removes his finger from the Beantable's surface.

Similarly, regarding physical object recognition, the **Physical Object and Cursor Recognition** service receives UDP data of the TUIO protocol specification about fiducials markers located on the Beantable's surface. Physical object information consists of: a) fiducial marker symbol id, b) position of the marker (two floating point values in a range of 0-1 representing a percentage scale within image's size), c) object state (i.e., FO_ADDED, FO_ACCELERATING, FO_DECELERATING, FO_STOPPED, FO_REMOVED, FO_ROTATING), and d) information about motion acceleration and rotation. The service offers to AmI clients the aforementioned physical object information by exposing the following functionality:

- **getPhysicalObjects**: returns to the AmI client a list (i.e. "**PhysicalObjectSeq**") of currently physical objects placed onto Beantable's surface.
- **Event_PhysicalObjectAdded:** notifies AmI clients when a new object is added (when a user places a physical object onto the Beantable's surface provided that it a fiducial marker at its abutting side).
- **Event_PhysicalObjectUpdated:** notifies AmI clients when an already tracked physical object is moved along the Beantable's surface.
- **Event_PhysicalObjectRemoved:** notifies AmI clients when an already tracked physical object is removed from the Beantable's surface.

Regarding motion acceleration and rotation, the following attributes are received from the reacTIVision and forwarded to AmI clients: a) the rotation acceleration value, i.e., "rotation_accel", b) the rotation speed value, i.e., "rotation_speed", c) the rotation angle value, i.e., "angle", d) the motion acceleration value, i.e., "motion_accel", e) the motion speed value, i.e., "motion_speed", f) the X-axis velocity value, i.e., "x_speed", and g) the Y-axis velocity value, i.e. "y_speed".

4.5.1.4 Force pressure sensitive interactive surface

Having a horizontally vision based-back projection interactive surface allows the easy placement of four force pressure sensitive resistors beneath it, as shown in Figure 58 (A, B). Each sensor varies its resistance depending on how much pressure is being applied to the sensing area. Despite the fact that these sensors are not extremely accurate, a software module uses them as a scale to capture information about the pressure exerted by the user during interactions. Every single sensor is attached to a FlexiForce adapter[57] which provides analog input to the Phidget Interface Kit[58], as shown in Figure 58 (C).

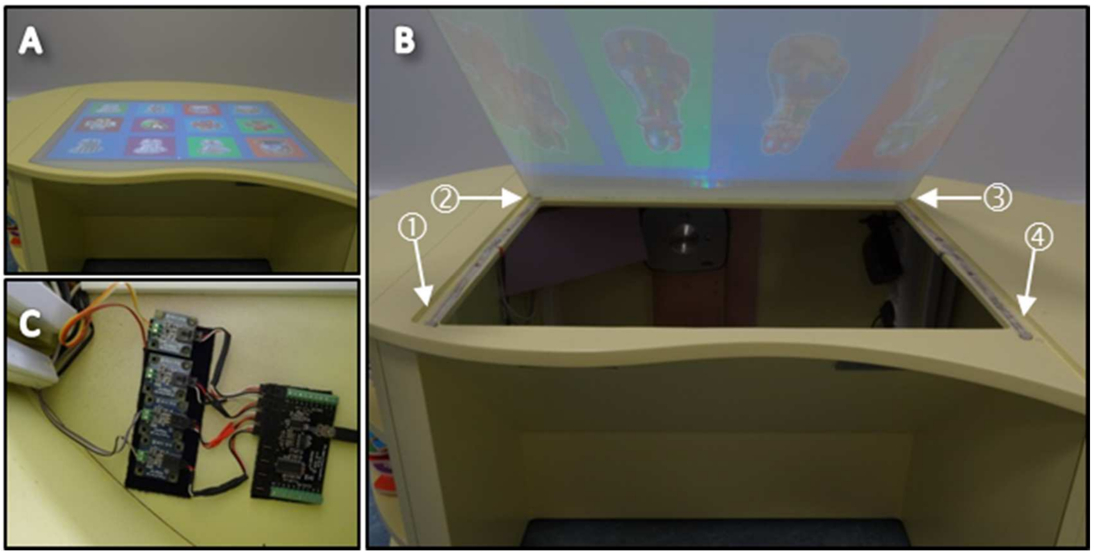


Figure 58: Deployment of force resistors to recognize pressure during interaction

The software module receives the analog values from the Phidget Interface Kit (which is connected via usb to the pc) and calculates the two-axis position of the exerted pressure using torque and rotational motion equations. Additionally, it measures the total force in pounds (lbs) by summing the applied force that each sensor receives. Because of the weight of the translucent Beantable's surface, as well as the sensor's inability to return always to the initial condition (as was before user's touching), the software module runs internally an auto-reset algorithm for calibration purposes. The latter ensures that whenever the interactive surface is empty (no touches, no physical objects, etc.), the captured data are used as a starting point for the next user interaction.

```
enum ForceAmountType { SOFT, MEDIUM, HARD};

struct DistributionInfo
{
       double x; //(0-1)
       double y; //(0-1)
       double force; // lbs
       long long time;
       ForceAmountType type;
};

struct SensorInfo
{
       long index;
       double force; //lbs
       long rawValue; //(0-1024)
       long long time;
};

//
// Methods
```

[57] http://www.phidgets.com/products.php?product_id=1120
[58] http://www.phidgets.com/products.php?product_id=1018

```
//
long GetCountOfSensors();
double GetValueForSensorWithIndex(in long index);
DistributionInfo GetForceDistributionInfo();

//
// Events
//
void Event_SensorChanged (in SensorInfo info);
void Event_DistributionChanged(in DistributionInfo info);
```

CodeBlock 5: Force pressure sensitive interactive surface. Service oriented implemented functionality (IDL)

Finally, the receiving end reports the outcomes to all the registered clients through the implemented IDL interface as depicted in CodeBlock 5. In particular, the service provides the clients with the following functionality:

- **GetCountOfSensors:** returns the number of the FlexiForce sensors placed beneath the translucent surface (for this setup it returns 4).
- **GetValueForSensorWithIndex:** returns the applied pressure measured in lbs for a specific force pressure sensor with index given as a parameter.
- **GetForceDistributionInfo:** returns information about the force pressure distribution. In particular, it returns information about the following: a) position in X, Y axes (two floating point values in a range of 0-1 representing a percentage scale within chair), i.e., “x”, “y”, b) the total applied force pressure (measured in lbs), i.e. ,“force”, c) a magnitude of the applied pressure (SOFT, MEDIUM, HARD), i.e., “type”, and d) the timestamp, i.e., “time”.
- **Event_SensorChanged:** notifies registered clients when the pressure of a sensor changes significantly. Moreover, it sends clients information about the following: a) index of the sensor whose value changed, i.e. “index”, b) the applied force pressure (measured in lbs), i.e. ,“force”, and c) the timestamp, i.e., “time”.
- **Event_DistributionChanged:** notifies registered clients when the force pressure distribution changes significantly, sending clients information as described in the case of **GetForceDistributionInfo.**

4.5.2 Augmented toys

Various toys have been augmented by placing special fiducial markers on their abutting side. For example, Figure 59 depicts the placement of fiducial marker on the bottom of each piece composing two jigsaw puzzles.

Figure 59: Jigsaw puzzle pieces' augmentation

Furthermore, a custom made stamp was developed to help very young children to perform traditional mouse clicks. Children can grab the stamp and "lift and hit" it on the surface (see Figure 60). This action is being interpreted as a touch event by the vision software subcomponents. This stamp seems to be very convenient for children who face difficulties in using multi-touch surfaces. It is worth mentioning that the vision software subsystem (i.e., reacTIVision) which interprets touch events was customized to recognize tiny blob areas which correspond to the touch of the children's small fingers on the surface.

Figure 60: The custom made stamp

4.5.3 Augmented pen

The augmented pen is a custom made pen in which all necessary hardware micro-electronic parts fit together while the measured dimensions do not exceed 2cm (L) X 2cm (W) X 16cm (H). The augmented pen was designed with SolidWorks[59] and came to reality via a 3D printer as depicted in Figure 61.

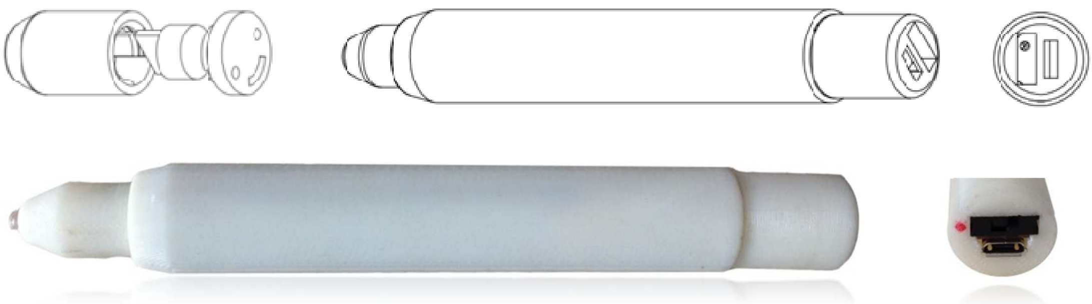


Figure 61: Augmented pen's SolidWorks design and the corresponded 3D printing

[59] http://www.solidworks.com/

As Figure 62 (A) depicts, various micro-electronic parts are used for measuring the applied pressure weight, the position and orientation on the screen, and the movement acceleration. In details, the pen consists of: a) an infrared led which lights automatically when the stylus is pressed on its tip against any surface, b) a force sensitive resistor in a specific embodiment in order to directly receive the pressure exerted by the user during writing, c) a small, thin, and low power triple axis accelerometer, d) a small microcontroller (i.e., Arduino Pro Mini[60]), and e) an RF link transmitter at 315Mhz. Embedding the aforementioned hardware infrastructure into the pen does not impact negatively its weight, which remains under 20 gr.

Figure 62: Augmented pen assembly

The augmented pen uses no ink but infrared light to make drawings. Due to the use of the infrared led, the augmented pen is suitable for interactive surfaces monitored by infrared spectrum cameras as in the case of Beantable. This setup allows pen's tip to be recognized as a cursor using **reacTIVision** as described in 4.5.1.3. The force sensitive resistor is attached to a custom made adapter which provides analog input to the microcontroller. The latter gathers input from every sensor and transmits them as raw data wirelessly to the RF receiver attached to another microcontroller connected to the computer, as shown in Figure 63.

Figure 63: Augmented pen data receiver

Moreover, an appropriate software module runs in the background in order to receive augmented pen's transmitted raw data. Such module computes the rotation matrix using

[60] http://arduino.cc/en/Main/ArduinoBoardProMini

the accelerometer values as well as the total force in Newton classifying the applied pressure as soft, normal or hard.

Augmented pen is powered with an ultra-small lithium rechargeable battery with a capacity of 40mAh and by utilizing the microcontroller's low power consumptions settings it can run continuously for almost a week. Nevertheless, the battery can be easily recharged via a micro usb socket placed beneath the cap as shown in Figure 62 (D).

```
enum ForceAmountType { SOFT, MEDIUM, HARD};
enum OrientationType { STANDING_UP, UPSIDE_DOWN, HORIZONTAL, OTHER};

struct SensorInfo
{
	double force; // in lbs
	double angle; // +90 means standing up,
	              // -90 means upside down,
	              // 0 means horizontal position

	ForceAmountType forceType;
	OrientationType orientType;

	long long time; //current time in msec

	long rawForceValue; // (0-1023)
	long rawAccXValue;  // (0-1023)
};

//
// Methods
//
long GetBatteryPercentage(); // 0 - 100, -1 means NO BATTERY connected
SensorInfo GetSensorInfo();
ForceAmountType GetTypeOfForceAmount();
OrientationType GetTypeOfOrientation();

//
// Events
//
void Event_SensorChanged (in SensorInfo info);
void Event_BatteryChanged(in long battPercentage);
```

CodeBlock 6: Augmented pen's service oriented implemented functionality (IDL)

Finally, the software module at the receiving end reports the outcomes to all the registered clients through the implemented IDL interface as depicted in CodeBlock 6. The service reads incoming data and provides the clients with the following functionality:

- **GetBatteryPercentage:** returns a value in the range 0-100 representing the percentage of the augmented pen's remaining battery.
- **GetTypeOfOrientation:** returns the relative with Beantable's surface orientation state (STANDING_UP, UPSIDE_DOWN, HORIZONTAL, OTHER).
- **GetTypeOfForceAmount:** returns the relative magnitude of the applied pressure (SOFT, MEDIUM, HARD).
- **GetSensorInfo:** returns information about: a) the applied pressure in the pen's tip measured in lbs, i.e., "force", b) the rotation of the pen in degrees, i.e. ,"angle", c) the

relative magnitude of the applied pressure, i.e., "forceType", d) the relative orientation state, i.e., "orientType", e) raw data about applied force pressure, i.e. "rawForceValue", and f) raw data about acceleration on X axis, i.e. "rawAccXValue".

- **Event_SensorChanged:** notifies registered clients when the applied force pressure on the pen's tip changes significantly. Moreover, it sends information as described in the case of **GetSensorInfo.**
- **Event_BatteryChanged:** notifies registered clients when the battery life measurement changes significantly.

4.5.4 Augmented chair

The augmeted chair was designed for young children (see Figure 64 (A)). It is a custom wooden prototype of the following dimensions: 68cm (L) X 51cm (W) X 78cm (H). The augmented chair can be used as an alternative medium for communication and multimodal interaction. As mentioned in **3.2.1.2**, its aim is twofold. Firstly, it can be used as a specialist joystick acting as an assistive technology pointing device. Secondly, it can be used to identify the psychological and physiological state of the children characterized by somatic and behavioral components such as hyperactivity.

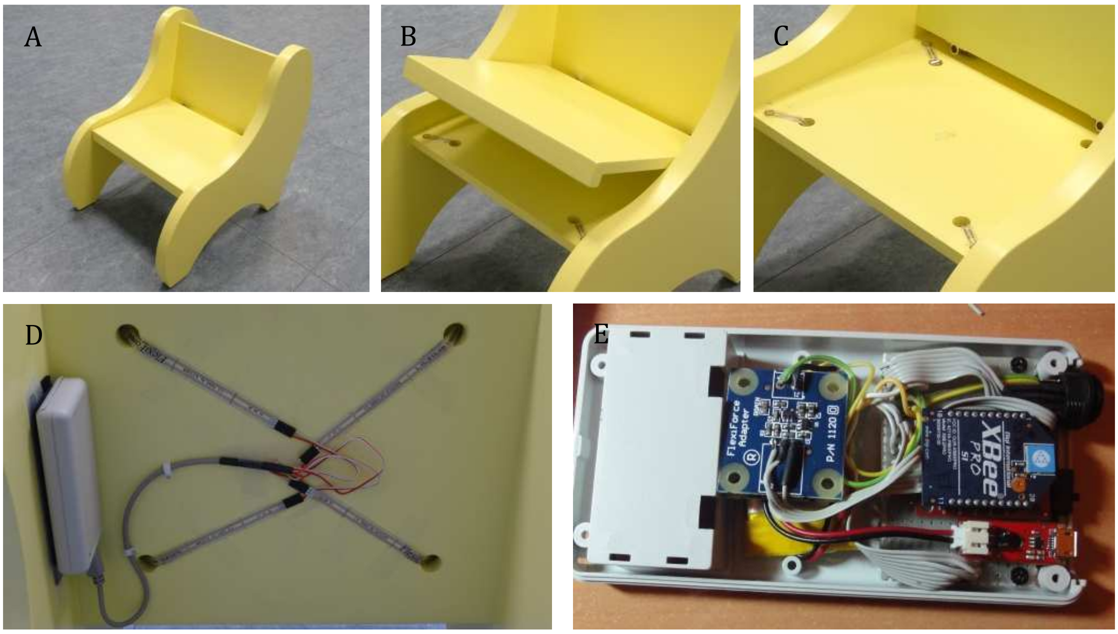


Figure 64: Augmented chair recognizes body posture data while sitting

The chair is able to capture body posture data using four FlexiForce 0-25lb resistive force pressure sensors[61] placed beneath the seat, as shown in Figure 64 (B, C). Every single sensor is attached to a FlexiForce adapter[62] which provides analog input to an XBee[63] module installed below the seat, as presented in Figure 64 (D, E). Thus, the chair transmits the captured data wirelessly to the pc where a software module receives them and calculates the two-axis position of the exerted pressure using torque and rotational

[61] http://www.phidgets.com/products.php?product_id=3101
[62] http://www.phidgets.com/products.php?product_id=1120
[63] http://en.wikipedia.org/wiki/XBee

motion equations. Additionally, it measures the total force in pounds[64] (lbs) by summing the applied force that each sensor receives. The XBee module is powered with a rechargeable battery of 3.7V with a large capacity of 2000mAh and is sufficient to run continuously for almost a couple of days. The battery can be easily recharged via a micro usb cable.

Figure 65: XBee explorer USB dongle.

```
enum ForceAmountType { SOFT, MEDIUM, HARD};

struct DistributionInfo
{
	double x; //(0-1)
	double y; //(0-1)
	double force; // lbs
	long long time;
	ForceAmountType type;
};

struct SensorInfo
{
	long index;
	double force; //lbs
	long rawValue; //(0-1024)
	long long time;
};

//
// Methods
//
long GetCountOfSensors();
double GetValueForSensorWithIndex(in long index);
DistributionInfo GetForceDistributionInfo();
long GetBatteryPercentage(); // 0 - 100, -1 means NO BATTERY

//
// Events
//
void Event_SensorChanged (in SensorInfo info);
void Event_DistributionChanged(in DistributionInfo info);
void Event_BatteryChanged(in long battPercentage);
```

CodeBlock 7: Augmented chair's service oriented implemented functionality (IDL)

Finally, in order for the augmented chair to receive and report the outcomes to all the registered clients, an IDL interface has been implemented as depicted in CodeBlock 7. In particular, the service communicates with XBee connected to the PC using an XBee

[64] http://en.wikipedia.org/wiki/Pound_%28mass%29

explorer dongle[65] (see Figure 65). The latter acts as a gateway between the computer and the XBee. The service reads incoming data and provides the clients with the following functionality:

- **GetCountOfSensors:** returns the number of the FlexiForce sensors placed beneath the seat. In case of augmented chair, it returns 4.
- **GetValueForSensorWithIndex:** returns the applied pressure measured in lbs for a specific force pressure sensor with index given as a parameter.
- **GetForceDistributionInfo:** returns information about the force pressure distribution. In particular, it returns information about the following: a) position in X, Y axes (two floating point values in a range of 0-1 representing a percentage scale within chair), i.e. "x", "y", b) the total applied force pressure (measured in lbs), i.e., "force", c) a magnitude of the applied pressure (SOFT, MEDIUM, HARD), i.e., "type", and d) the timestamp, i.e., "time".
- **GetBatteryPercentage:** returns a value in the range 0-100 representing the percentage of the remaining battery.
- **Event_SensorChanged:** notifies registered clients when the pressure of a sensor changes significantly. Moreover, it sends clients information about the following: a) index of the sensor whose value changed, i.e., "index", b) the applied force pressure (measured in lbs), i.e., "force", and c) the timestamp, i.e., "time".
- **Event_DistributionChanged:** notifies registered clients when the force pressure distribution changes significantly. In detail, it sends clients information as described in the case of **GetForceDistributionInfo.**
- **Event_BatteryChanged:** notifies registered clients when the battery life measurement changes significantly.

4.5.5 Digital dice

In terms of building an augmented dice, the embedded microelectronics approach was chosen over the visual one, as further described below (see 3.2.1.5.2). To this end, DICE+[66] which is an interactive gaming dice already available on the market was selected. The choice of DICE+ was made to give the children the affordance of using traditional-looking dice while supporting the wireless communication of the rolling results to the registered clients. DICE+ is compatible with almost any mobile device with Bluetooth 2.1 and 4.0 such as Android and iOS devices.

DICE+ provides: a) an accelerometer with the magnetic field sensor resulting in 3D orientation, b) 6 independent touch sensitive faces, c) 6 independent LEDs that can glow in any color, d) a battery that gives 20h of game playing, e) a micro-USB for easy charging, and f) a thermometer.

[65] https://www.sparkfun.com/products/11697
[66] http://dicepl.us/

Figure 66: DICE+

4.5.5.1 Digital dice service implementation details

Due to the fact that DICE+ communicates only with mobile devices, an iOS device (i.e., iPad) was selected to host a service application acting as the mediator between the AmI environment and the dice. The implemented iOS employs a middleware network layer in order to expose dice functionality to the network clients. The middleware service is comprised of different modules that are responsible for handling dices (e.g., dice discovery, connection, etc.) as well as for receiving and forwarding embedded sensors' captured data. These modules are described in detail in the following sections. To save battery power, DICE+ does not send captured data via Bluetooth when it is connected. Clients need to specify what notifications they want to use by enabling/starting the corresponding module.

4.5.5.1.1 Roll

The most obvious sensor is the roll result readout. This is a complex sensor that not only tells the mediator what number was rolled on the dice but also takes care of any problems that might have occurred while the dice was rolling. When DICE+ is rolled, the service application receives not only the result but also the duration of the roll and its status describing if the roll was proper or not. Clients can stop roll updates in order to save energy. Exposed functionality is described in the interface definition language (IDL) as follows:

```
enum RollFlags { RollFlagOK, RollFlagTilt, RollFlagTooShort};
void StartRollUpdates();
void StopRollUpdates();
void Event_DidRoll (in long result, in RollFlags flags, in long duration);
```

CodeBlock 8: Digital dice exposed functionality for rolling sensor (IDL)

4.5.5.1.2 Accelerometer

This sensor can detect the slightest movement of the controller. For example, captured data can be used for detecting gestures or control a magic wand. The service gets notified about changes to accelerometer readouts including four values: a timestamp and one accelerometer readout (measured in milli Gs) for one of the axis X, Y, Z. Clients can stop accelerometer updates in order to save more energy. Exposed functionality is described in the interface definition language (IDL) as follows:

```
void StartAccelerometerUpdates();
void StopAccelerometerUpdates();
void Event_DidAccelerate(in long x, in long y, in long z);

void StartTapUpdates();
void StopTapUpdates();
void Event_DidTap(in long x, in long y, in long z);
```

CodeBlock 9: Digital dice exposed functionality for accelerometer sensor (IDL)

4.5.5.1.3 Magnetometer

Magnetometer can detect moves that do not get handled by the accelerometer, for example a slow roll around one of the axis. The service gets notified about changes to magnetometer readouts including four values: a timestamp and one magnetometer readout (measured in micro Teslas) for each of the axis. Clients can stop magnetometer updates in order to save energy. Exposed functionality is described in the interface definition language (IDL) as follows:

```
void StartMagnetometerUpdates();
void StopMagnetometerUpdates();
void Event_DidUpdateMagnetometer(in long x, in long y, in long z);
```

CodeBlock 10: Digital dice exposed functionality for magnetometer sensor (IDL)

4.5.5.1.4 Orientation

This service can read a full 3D orientation of DICE+ by combining readouts from accelerometer and magnetometer. The service gets notified about changes to orientation readouts including four values: a timestamp and yaw, pitch and roll values which indicate rotation (measured in degrees) around each of the axis. Similarly, the service gets notified with the number of the face which is currently pointing up. Clients can stop orientation or face change updates in order to save energy. Exposed functionality is described in the interface definition language (IDL) as follows:

```
void StartOrientationUpdates();
void StopOrientationUpdates();
void Event_DidUpdateOrientation(in long yaw, in long pitch, in long roll);

void StartFaceChangeUpdates();
void StopFaceChangeUpdates();
void Event_DidChangeFace(in long face);
```

CodeBlock 11: Digital dice exposed functionality for orientation sensor (IDL)

4.5.5.1.5 Temperature

According to DICE+ specifications, the magnetic field sensor has a temperature sensor. The service gets notified about changes to thermometer readouts including two values:

a timestamp and the temperature value (measured in Celsius). Clients can stop temperature updates in order to save energy. Exposed functionality is described in the interface definition language (IDL) as follows:

```
void StartThermometerUpdates();
void StopThermometerUpdates();
void Event_DidUpdateTemperature(in float temperature);
```

CodeBlock 12: Digital dice exposed functionality about temperature sensor (IDL)

4.5.5.1.6 Touch

Every side on DICE+ is touch sensitive, making it possible to understand if someone is holding it. For example, DICE+ can be used as a 6 digit keyboard, making it possible to select elements in the game environment. The service gets notified about changes to touch readouts including three values: a) a timestamp at which the measurement was taken, b) a bit mask indicating which faces are barrenly being touched, and c) a change bit mask indicating which faces have changed their state between this and previous readout. The touch sensor is an analog one, which means that the service is not only able to get notified when a face is being touched but also to measure the touch's "strength". The service gets notified about changes to proximity readouts they include two values: a timestamp and an array of proximity values for each of the faces. Clients can stop touch or proximity updates in order to save energy. Exposed functionality is described in the interface definition language (IDL) as follows:

```
void StartTouchUpdates();
void StopTouchUpdates();
void Event_DidUpdateTouches(in long changeMask, in long currentStateMask);

void StartProximityUpdates();
void StopProximityUpdates();
void Event_DidUpdateProximity(in ami::LongSeq values);
```

CodeBlock 13: Digital dice exposed functionality for touch sensors (IDL)

4.5.5.1.7 LEDs

With full access to the LEDs clients can control which faces need to (should) light up, what should the color be and what animation should it play. Clients (e.g., smart games) can use these LEDs to inform the user about whose turn is it or when the game status changed, etc. Each face of the dice contains an RGB LED. Clients can control the LEDs using animations.

Dice+ supports three types of animations that can be started by a client application: a) blink animations, b) fade animations, and c) built-in animations. For the first two animation types, clients can select: a) which faces should be animated, b) what should their color be, c) how long should the animation last, d) how many times it should be repeated, e) animation's priority (the lower the value the more important the animation is, sending an animation with priority 0 stops the currently playing animation and resumes it after the new one is done). The built-in animations allow clients to play one of the four "standard" notifications that the dice uses for: a) valid rolls, b) invalid roll, c) wake-up, and d) shutdown. Exposed functionality is described in the interface definition language (IDL) as follows:

```
void StartBlinkAnimationWithMask(in long ledMask, in long priority, in long
r, in long g, in long b, in long onPeriod, in long cyclePeriod, in long
blinkCount);

void StartFadeAnimationWithMask(in long ledMask, in long priority, in long
r, in long g, in long b, in long fadeTime, in long pauseTime);

void StartFadeAnimationWithMaskExt(in long ledMask, in long priority, in
long r, in long g, in long b, in long fadeTime, in long pauseTime, in long
repeatCount);
```

CodeBlock 14: Digital dice exposed functionality for leds control (IDL)

4.5.5.1.8 Battery

According to DICE+ specifications[67], the embedded battery gives up to 20h of game playing and 3 months of standby. Service gets notified about battery state including the current battery level and its status (whether it's charging or not). For example, clients inform user in case that the die has low battery and should be connected to a power source. Battery level readouts they include two values: a timestamp and the current battery level. Similarly, battery status readouts include two values: a timestamp and a flag indicating whether the die is charging. Clients can stop battery updates in order to save energy. Exposed functionality is described in the interface definition language (IDL) as follows:

```
void StartBatteryUpdates();
void StopBatteryUpdates();
void Event_DidUpdateBatteryStatus(in long level, in boolean isCharging, in
boolean isLow);
```

CodeBlock 15: Digital dice exposed functionality for battery updates (IDL)

4.5.5.1.9 Power mode

Because the fact that DICE+ is powered by a battery, it automatically sets itself in sleep mode for battery saving purposes. The dice sends power mode notification about power events. The service gets notified when the dice wants to turn into sleep mode, and forwards the notification to the registered clients which can subsequently inform the user about that. The receiving end receives and forwards to registered clients power mode readouts, which include two values: a timestamp and a flag indicating the current power mode. The flag can have 1 of three values representing: a) whether the die is operating normally, b) whether the die is shutting down, and c) whether the die is asleep. Clients can stop power mode updates in order to save energy. Exposed functionality is described in the interface definition language (IDL) as follows:

```
enum PowerMode { Operating, GoingToSleep, Sleeping};

void StartPowerModeUpdates();
```

[67] http://dicepl.us/

```
void StopPowerModeUpdates();
void Event_DidUpdatePowerMode(in PowerMode mode);
```

CodeBlock 16: Digital dice exposed functionality for power mode events (IDL)

4.5.5.1.10 Connection management

This mobile service allows clients to scan for available dices. When a client starts scanning, the service initiates connection to an available die. The service sends notification every time a die is being connected or disconnected. Exposed functionality is described in the interface definition language (IDL) as follows:

```
void StartScan();
void StopScan();
boolean IsDiceConnected();

void Event_DiceConnected();
void Event_DiceDisconnected();
void Event_DiceManagerStoppedScan();
```

CodeBlock 17: Digital dice exposed functionality for dice discovery, connection and disconnection (IDL)

4.6 Text to Speech services

In order to employ verbal communication channels between the envisioned smart games and the children, the proposed framework includes a generic solution that is able to convert text into computer-generated voice output. To this end, a set of text to speech (TTS) services were developed. This set includes: a) a **Sound Retouch** service, and b) a **Text to Speech** service. Both services are hosted by a windows server that integrates the Microsoft Speech API (SAPI). As a result, **Text to Speech** service provides access to the functionality of an installed speech synthesis engine and therefore is able to turn requested text into speech using various free or commercial TTS products like Acapela[68] or Loquendo[69]. Through middleware the text to speech service accepts requests from any remote client application, i.e., a smart game, and returns back (through the network) the speech audio stream accompanied with the phonemes and visemes reached. Exposed **Text to Speech** service functionality is described in the interface definition language (IDL) as follows:

```
enum VoiceAge { VoiceAgeNotSet, Child, Teen, Adult, Senior };
enum VoiceGender { VoiceGenderNotSet, Male, Female, Neutral };

struct AdditionalInfo
{
	string key;
	string value;
};

typedef sequence<AdditionalInfo> AdditionalInfoSeq;

struct VoiceInfo
{
```

[68] http://www.acapela-group.com
[69] http://www.nuance.com

```
        AdditionalInfoSeq   AdditionalInfo;
        VoiceAge Age;
        string CultureInfo;
        string Description;
        VoiceGender Gender;
        string Id;
        string Name;
};

typedef sequence<VoiceInfo> VoiceInfoSeq;

enum SynthesizerEmphasis { Stressed, Emphasized, None };

struct VisemeReached
{
        unsigned long long AudioPosition;
        unsigned long long Duration;
        SynthesizerEmphasis Emphasis;
        long NextViseme;
        long Viseme;
};

typedef sequence<VisemeReached> VisemeReachedSeq;

struct PhonemeReached
{
        unsigned long long AudioPosition;
        unsigned long long Duration;
        SynthesizerEmphasis Emphasis;
        wstring NextPhoneme;
        wstring Phoneme;
};

typedef sequence<PhonemeReached> PhonemeReachedSeq;

enum Mode { NORMAL, VISEMES, PHONEMES, ALL };

struct VoiceObject
{
        ami::OctetSeq AudioStream;
        VisemeReachedSeq Visemes;
        PhonemeReachedSeq Phonemes;
};

VoiceInfoSeq GetInstalledVoices();

VoiceObject Speak(in wstring  text, in string voicename, in Mode mode, out
string error);

VoiceObject SpeakAtVolume(in wstring  text, in string voicename, in long
volume, in Mode mode, out string error);

VoiceObject SpeakAtRate(in wstring  text, in string voicename, in long rate,
in Mode mode, out string error);

VoiceObject SpeakAtVolumeAndAtRate(in wstring  text, in string voicename, in
long volume, in long rate, in Mode mode, out string error);

boolean TrySpeak(in wstring  text, in string voicename, in Mode mode, out
VoiceObject voiceObj, out string error);

boolean TrySpeakAtVolume(in wstring  text, in string voicename, in long
volume, in Mode mode, out VoiceObject voiceObj, out string error);

boolean TrySpeakAtRate(in wstring  text, in string voicename, in long rate,
in Mode mode, out VoiceObject voiceObj, out string error);

boolean TrySpeakAtVolumeAndAtRate(in wstring  text, in string voicename, in
```

```
long volume, in long rate, in Mode mode, out VoiceObject voiceObj, out
string error);

VoiceObject SpeakSsmlMarkup(in wstring  ssmlMarkup, in Mode mode, out string
error);

boolean TrySpeakSsmlMarkup(in wstring  ssmlMarkup, in Mode mode, out
VoiceObject voiceObj, out string error);

boolean SaveVoiceObject(in VoiceObject voiceObject, in string path, in
string id);
```

CodeBlock 18: Text to Speech service oriented functionality (IDL)

The service is implemented in C# using Microsoft Speech Platform SDK[70] and provides clients with the following functionality:

- **GetInstalledVoices**: returns information about the installed voices of the server that hosts TTS products. Each voice is mainly characterized by the following: a) age of the voice, i.e. "Age", b) culture, i.e., *"CultureInfo"*, c) description, i.e., *"Description"*, d) gender *(VoiceGenderNotSet, Male, Female, Neutral)*, i.e. *"Gender"*, e) an identification number, i.e., *"Id"*, and f) name of the voice, i.e. *"Name"*.
- **Speak**: converts text given as a parameter, i.e., *"text",* using selected voice, i.e., *"voicename"* to sound. The returned value, i.e., *"VoiceObject"* consists of the computer-generated voice output, i.e., "*AudioStream*", accompanied by a list of reached visemes or/and phonemes according to the selected mode, i.e., *"mode".* If selected mode is *NORMAL*, neither visemes nor phonemes will be reached. Similarly, in case of ALL, both visemes and phonemes will be reached. Each reached viseme, i.e., *"VisemeReached"* contains information about the: a) position of the viseme in the audio stream in milliseconds, i.e., *"AudioPosition"*, b) duration of the viseme in millieseconds, i.e. *"Duration"*, c) a magnitude of the emphasis *(Stressed, Emphasized, None)* that describes the viseme, i.e., *"Emphasis"*, d) next viseme id, i.e., *"NextViseme"*, and e) the id of the current viseme, i.e., *"Viseme"*. Similarly, each reached phoneme, i.e., *"PhonemeReached"* contains information about the: a) position of the phoneme in the audio stream in milliseconds, i.e., *"AudioPosition"*, b) duration of the phoneme in millieseconds, i.e., *"Duration"*, c) a magnitude of the emphasis *(Stressed, Emphasized, None)* that describes the phoneme, i.e., *"Emphasis"*, d) next phoneme id, i.e. *"NextPhoneme"*, and e) the id of the current phoneme, i.e., *"Phoneme".*
- **SpeakAtVolume**: same functionality as **Speak** with the difference that the client can adjust the volume of the generated voice by setting the parameter *"volume"* accordingly.
- **SpeakAtRate**: same functionality as **Speak** except that the client can set the speaking rate of the generated voice by setting the parameter *"rate"* accordingly.
- **SpeakAtVolumeAndAtRate**: merges functionality of SpeakAtVolume and SpeakAtRate.
- **TrySpeak**: same functionality as **Speak** with the difference that in case of service fails to convert text to voice it returns *FALSE,* otherwise it returns *TRUE.* The result of the computer generated voice is returned via the out parameter "*voiceObj*".

[70] https://msdn.microsoft.com/en-us/library/hh362831(v=office.14).aspx

- **TrySpeakAtVolume**: same functionality as **TrySpeak** with the difference that the client can adjust the volume of the generated voice by setting the parameter *"volume"* accordingly.
- **TrySpeakAtRate**: same functionality as **TrySpeak** except that the client can set the speaking rate of the generated voice by setting the parameter *"rate"* accordingly.
- **TrySpeakAtVolumeAndAtRate**: merges functionality of TrySpeakAtVolume and TrySpeakAtRate.
- **SpeakSsmlMarkup**: same functionality as **Speak** with the difference that the text given as a parameter, i.e., *"ssmlMarkup"*, is an XML-based markup language that clients use to control various characteristics of synthetic speech output including voice, pitch, rate, volume, pronunciation, and other characteristics[71].
- **TrySpeakSsmlMarkup**: same functionality as **SpeakSsmlMarkup** with the difference that in case of service fails to execute it returns *FALSE*, otherwise it returns *TRUE*. The result of the computer generated voice is returned via the out parameter "*voiceObj*".
- **SaveVoiceObject**: stores the voice object given as a parameter, i.e., *"VoiceObject"* to a specific path, i.e., *"path"* with filename based on selected id, i.e., *"id"*. In detail, the following files are being generated: a) the audio stream with filename *id*.wav, b) a text file containing the reached visemes using the format *AudioPosition* \ *Duration* \ *Emphasis* \ *NextViseme* \ *Viseme* and saved with the filename *id*_visemes.txt, and c) a text file containing the reached phonemes using the format *AudioPosition* \ *Duration* \ *Emphasis* \ *NextPhoneme* \ *Phoneme* and saved with the filename *id*_phonemes.txt.

The **Sound Retouch** service can be used to change the tempo, pitch and rate of the speech audio stream received as input to match custom defined speaker characteristics, i.e., the voice age. Exposed sound retouch service functionality is described in the interface definition language (IDL) as follows:

```
ami::OctetSeq ProcessWave (in ami::OctetSeq audioStream, in float newTempo,
in float newPitch, in float newRate, out string error);

void ProcessWaveOnRemoteFolder (in string fileIn, in string fileOut, in
float newTempo, in float newPitch, in float newRate, out string error);
```

CodeBlock 19: Sound Retouch service oriented functionality (IDL)

The service is implemented in C++ using the SoundTouch[72] audio processing open-source library provides clients with the following functionality:

- **ProcessWave:** returns a modified version of the input audio stream, i.e., "audioStream" according to: a) the tempo given as a parameter, i.e., *"newTempo"*, b) the pitch given as a parameter, i.e., *"newPitch"*, and c) the rate given as a parameter, i.e., "newRate". Tempo (time strength) changes the sound to play at faster or slower tempo than originally without affecting the sound pitch (tempo value equal to 1 has

[71] https://msdn.microsoft.com/en-us/library/hh378377
[72] http://www.surina.net/soundtouch/

no effect). Pitch changes the sound's musical tome without changing the original tempo (pitch value equal to 1 has no effect). Rate (playback rate) changes both tempo and pitch as if a vinyl disc was played at different speed (rate value equal to 1 has no effect).

- **ProcessWaveOnRemoteFolder:** provides the same functionality as ProcessWave except that the service receives as a parameter the file location of an audio stream, i.e., *"fileIn"*, in order to read and process according to the remaining parameters and save the modified version in a file location given as a parameter, i.e., *"fileOut"*.

4.7 Max: A cross-platform, remotely-controlled mobile avatar simulation framework as a play-fellow

In the developed framework, a virtual character was envisioned to act as the child's virtual partner facilitating natural interaction with young children. This section presents the implementation details of a novel, remotely controlled three-dimensional full body avatar simulation framework.

Figure 67: The cross platform three-dimensional avatar

Max is a remote controlled full body three-dimensional avatar that supports multi-presence almost in any available device in an AmI environment (see Figure 67). The role of Max depends on the client-application's requirements. Typical examples include acting as a guide, assistant or information presenter. In order to achieve natural communication channels both non-verbal and verbal behavior is essential. Non-verbal communication includes full body animation and facial expressions. For example, when idle, the virtual character is never motion-less due to an undulating body animation and eyes blinks randomly, giving the illusion that it is alive. The facial animation is strengthened by raising the eyebrows. Max can also present multimedia content on the television contained in the scene.

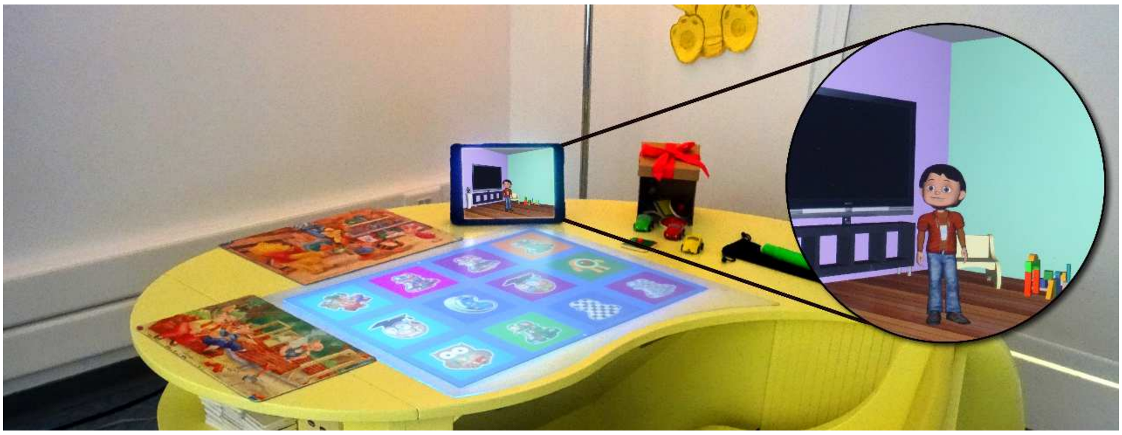


Figure 68: Max acting as virtual partner for the needs of an augmented interactive table for young children

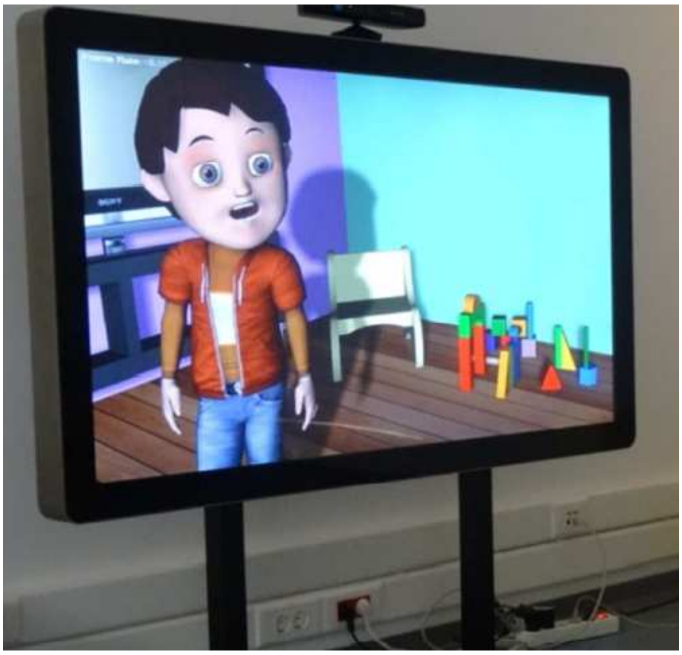


Figure 69: Max in the Mimesis game asks the child to imitate body postures

As depicted in Figure 68, Max accompanies Beantable by running on the secondary display device as presented in 3.2.1.3. To this end, an iOS mobile device (iPad), is used to present the child's virtual partner supporting various modes of interaction such as playful or didactic. Each smart game requests Max to act as a friend or opponent of the child, giving instructions or insisting on the completion of a task. Additionally, Max employs a large display to create the so called "Mimesis game" (see Figure 69). In this smart game (presented analytically in the following section 5.1.3), Max is presented in the large display that integrates a Windows 7 operating system and interacts with the child in a very natural way using both verbal and non-verbal communication channels. More specifically, Max requests the child to imitate the various body postures he makes. Using Nibbler, the sensory infrastructure presented in 4.4, the smart game measures the quality and performance of the body posture that the child assumes and extracts indications regarding the archived maturity level and skills of the child.

4.7.1 Architecture of Max

The novel AmI architecture presented in this section mainly focuses on the development of a uniquely cross-platform, mobile three-dimensional avatar that could be remotely controlled by any client application in the context of an AmI environment (e.g., smart game). This is carried out through a networking middleware [76]. As depicted in Figure 70, Max accepts remote procedure calls from diverse clients (e.g., smart games) to animate, read some text, assume a posture, generate a specific facial expression, or present multimedia content. As shown in Figure 70, these functions are served by the following software components: a) Camera Path Animations, b) Skeleton Animations, c) Text to Speech, d) Media Presentation, and e) Facial Expressions.

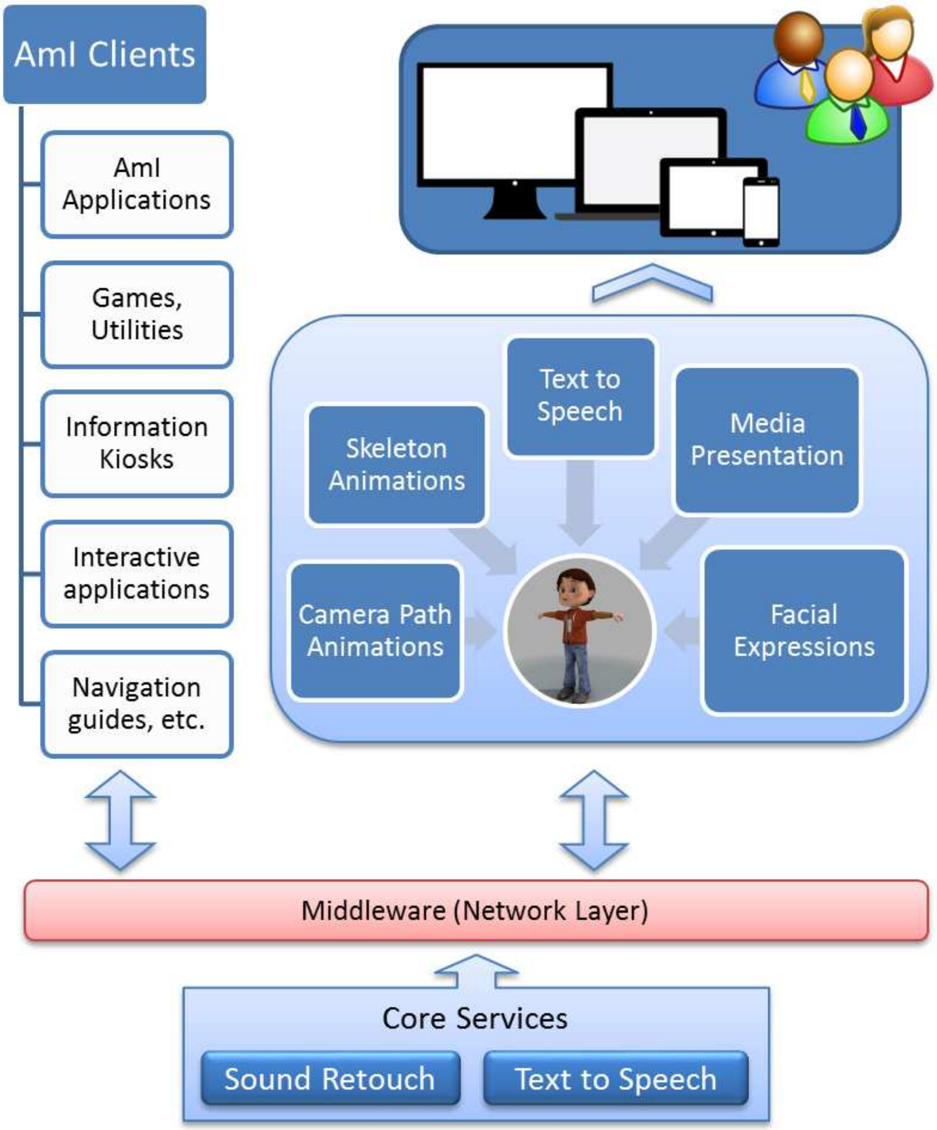


Figure 70: Architecture of Max: A three-dimensional remotely controlled cross-platform avatar

As mentioned before, Max enhances natural communication using verbal communication channels. Because there is no yet available a cross platform text to speech (TTS) system capable to run in any device or platform, Max adopts the aforementioned Text to Speech services as presented in 4.6. Thus, Max overcomes this difficulty and is able to talk regardless the selected device or platform. Through the middleware, the **Text to Speech** service accepts requests from Max, and returns back (through the network) the speech audio stream accompanied with the phonemes and visemes reached. If necessary, the sound retouch service can be used to change the tempo, pitch and rate of the speech audio stream received as input to match custom defined speaker characteristics, i.e., the voice age.

4.7.2 Max implementation details

Max is a six year old low polygon three-dimensional character (i.e. 26K polygons) which originally was purchased from Turbosquid and modified in order to achieve increased rendering performance (reducing the number of polygons to 23K). The model was already rigged with biped system and was animation ready including some facial expressions. Minor changes were necessary in the context of skinning and texturing.

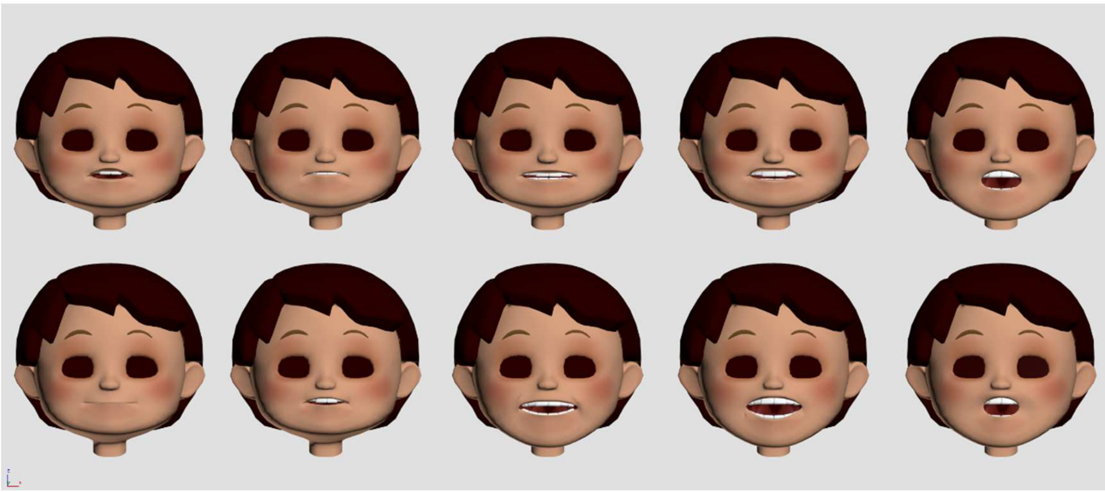


Figure 71: Implemented morph targets

Now, Max is available in Collada, which is an open standard XML schema for exchanging digital assets among various graphics software applications. Max "came to life" using OpenSceneGraph 3.0.1, an open source cross platform 3D graphics application programming interface. OpenSceneGraph is used by application developers in fields such as visual simulation, computer games, virtual reality, scientific visualization and modelling. Max is currently built for the Apple's mobile operating system (iOS) and for the Windows operating system, as well as tested in iPhone, iPad and PC. The frame rate in mobile devices was measured at 24 fps without shadows and at 60 fps with shadows enabled in PC using an ordinary graphics card. According to the aforementioned architecture, Max is comprised of different modules that are responsible for executing remote requests received from client applications. These modules are described in the following sections.

4.7.2.1 Text to Speech

Max supports lip-synchronization using the **Text to Speech** service. Specifically, when Max receives some text to read, he uses the **Text to Speech** service (via middleware) to get back the equivalent audio stream accompanied with a list of the reached visemes. According to the request's requirements, Max may submit the audio stream to the **Sound Retouch** service to change voice characteristics i.e., voice tone, pitch or tempo. Such list contains information about each reached viseme in the following format: *AudioPosition**Duration**Emphasis**NextViseme**Viseme*.

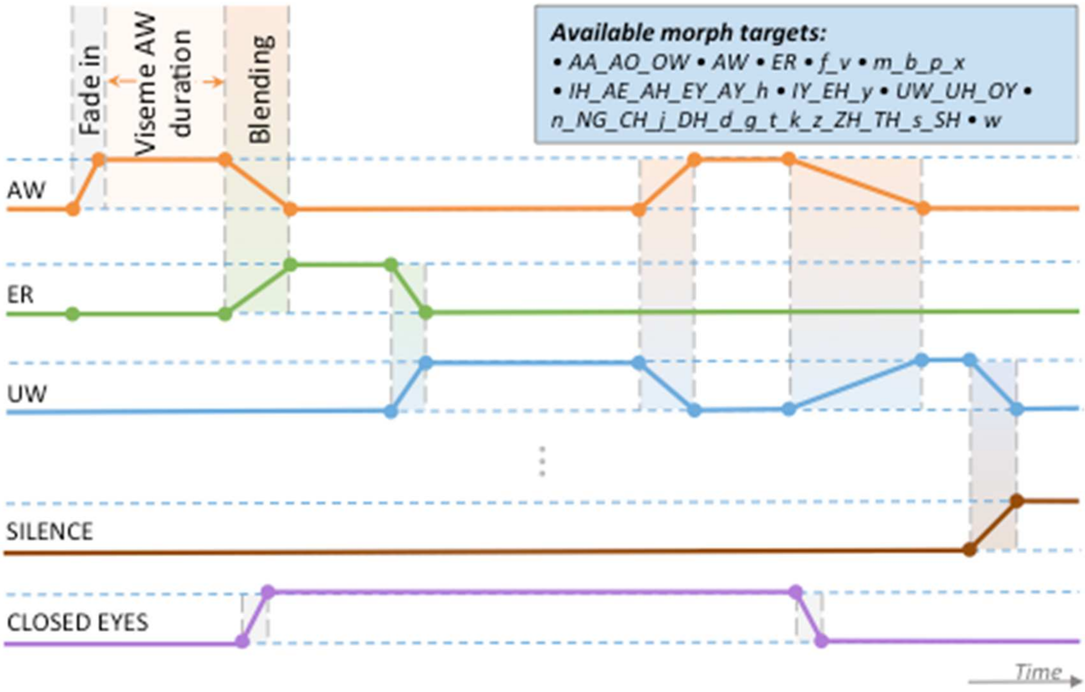


Figure 72: Blending formula among various animation channels for each morph target.

Given that information, before starting reading, Max prepares an animation consisting of various float linear animation channels. As shown in Figure 72, each animation channel smoothly activates or deactivates the morph target corresponding to each viseme contained in the visemes list. In detail, a morph target begins to fade in from the previous viseme's audio position added by its duration and fades out until the audio position of the next one. Random eyes blinking is implemented in the same way. Various tests indicated that this technique has better results even if some animation key-frames are lost due to rendering instability. Every received request can be identified by a unique keyname. The purpose of the latter is to avoid unnecessary requests to the text to speech and sound retouch service. When TTS data already exist in Max's documents folder, they are recycled to reduce network traffic unless client forces cache replacement. Max sends an event to the remote client in order to notify that reading has finished. Exposed Text to Speech's functionality is described in the interface definition language (IDL) as follows:

```
// @param voicename: Sets the tts voice i.e. Afroditi
// @param speakRate: Sets the speaking rate from -10 through 10 (default 0).
// @param volume: Sets the output volume from 0 through 100 (default 50).
// @param tempo (time stretch): Changes the sound to play at faster or
slower tempo than originally without affecting the sound pitch (default 1.0)
// @param pitch (key): Changes the sound pitch or key while keeping the
```

```
original tempo (speed) (default 1.0)
// @param soundRate: Changes both tempo and pitch together as if a vinyl
disc was played at different RPM rate (default 1.0)

ami::StringSeq GetAvailableText2SpeechData();

boolean Speak(in string keyname, out string error);

boolean SpeakText(in wstring text, in string voicename, in string keyname,
in boolean replaceCached, out string error);

boolean SpeakTextExt(in wstring  text, in string voicename, in string
keyname, in long volume, in long speakRate, in float tempo, in float pitch,
in float soundRate, in boolean replaceCached, out string error);

boolean AddText2SpeechData(in string keyname, in ami::OctetSeq audioStream,
in ami::OctetSeq visemesStream, in ami::OctetSeq phonemesStream, in boolean
forceOverwrite, out string error);

void Event_Text2SpeechAnimationCompleted (in string keyname);
```

CodeBlock 20: Text to Speech's service oriented exposed functionality (IDL)

4.7.2.2 Facial Expressions

Max shows emotions through facial expressions. Typical examples of morph targets used in facial animation include a smiling mouth, a closed eye, and a raised eyebrow. Apart from making a facial expression using only a morph target at a time, Max is able to animate among different facial expressions in a timeline. To this end, the facial expression manager adopts the same blending functionality as described in the previous section to animate among different morph targets. Additionally, a remote client can submit a 3D model in .dae format that contains a morph geometry which corresponds to a new facial expression. Max sends an event back to the remote clients in order to notify that a facial expression or animation has completed. Exposed Facial Expressions' functionality is described in IDL as follows:

```
struct FacialExpressionEvent
{
       double time;
       string expression;
};

typedef sequence<FacialExpressionEvent> FacialExpressionEventSeq;

ami::StringSeq GetAvailableFacialExpressions();

boolean PlayFacialExpression(in string keyname, in long duration, out string
error);

boolean PlayFacialAnimation(in string keyname, in FacialExpressionEventSeq
expressions, out string error);

boolean AddFacialExpression(in string keyname, in ami::OctetSeq morphStream,
in boolean forceOverwrite, out string error);

void Event_FacialExpressionCompleted (in string keyname);
void Event_FacialAnimationCompleted (in string keyname);
```

CodeBlock 21: Facial Expressions' service oriented exposed functionality (IDL)

4.7.2.3 Camera Path Animations

The camera's translation and rotation matrix can change dynamically depending on the needs of remote clients. For example, during multimedia presentation, the camera moves to focus the television. As a result, Max can present multimedia content in full screen mode on the running device. To this end, Camera Path Animation module is responsible to animate the camera's position and orientation according to the path received remotely from the client application. Additionally, it can provide information about the current's camera position and orientation, as well as receive a text file that contains a timeline of paths. Each timeline is identified and stored locally by its unique keyname for caching. Max sends an event to the remote clients in order to notify that a camera path animation has been completed. Exposed Camera Path Animations module's functionality is described in IDL as follows:

```
ami::StringSeq GetAvailableCameraPathAnimations();

boolean PlayCameraPathAnimation(in string keyname, out string error);

boolean AddCameraPathAnimation(in string keyname, in ami::OctetSeq
animationStream, in boolean forceOverwrite, out string error);

string GetCurrentCameraPosition();

void Event_CameraPathAnimationCompleted (in string keyname);
```

CodeBlock 22: Camera Path Animations module's service oriented exposed functionality (IDL)

4.7.2.4 Media Presentation

Multimedia content is presented on the television next to Max. The Media Presentation module uses ffmpeg[73] to open and play almost any known format of videos or images while audio playback is currently supported only for .wav audio streams. In order to reduce network traffic, the Media Presentation module is able to receive and store locally a multimedia file and play it when needed. Furthermore, it sends an event to notify that a multimedia content presentation has ended. In case of pictures, that event is sent right after the appearance of the image. Exposed Media Presentation's functionality is described in IDL as follows:

```
ami::StringSeq GetAvailableMediaFiles();

boolean ShowMedia(in string filename, out string error);

boolean AddMedia(in string filename, in ami::OctetSeq mediaStream, in
boolean forceOverwrite, out string error);

void Event_MediaPresentationCompleted (in string filename);
```

CodeBlock 23: Media Presentation's service oriented exposed functionality (IDL)

[73] https://www.ffmpeg.org/

4.7.2.5 Skeleton Animations

Max implements body animation through biped skeletal animation using key-frame interpolation. Typical examples of biped animations include walking, clapping, running, dancing and bowing. Remote clients may receive a list of available animations or add a new one by submitting a 3D model in .dae format containing an animation. Clients can also set the number of loops as well as the playing mode (once, stay, loop or ping pong). The Skeleton Animation module notifies remote clients when the animation has finished. Exposed Skeleton Animation's functionality is described in IDL as follows:

```
enum PlayMode { ONCE, STAY, LOOP, PPONG };

ami::StringSeq GetAvailableSkeletonAnimations();

boolean PlaySkeletonAnimation(in string keyname, in long loops, in PlayMode
playMode, in long startTime, out string error);

boolean AddSkeletonAnimation(in string keyname, in ami::OctetSeq
animationStream, in boolean forceOverwrite, out string error);

void Event_SkeletonAnimationCompleted (in string keyname);
```

CodeBlock 24: Skeleton Animation's service oriented exposed functionality (IDL)

4.8 ACTA: A general purpose finite state machine (FSM) description language for ACTivity Analysis

This section presents ACTA, a general purpose finite state machine (FSM) description language for rapid prototyping of smart games. ACTA facilitates the activity analysis process during smart game design by early intervention professionals who are not familiar with traditional programming languages. Furthermore, smart game developers can use ACTA not only for developing event-driven sequential logic games, but also for applications of behavior composed of a finite number of states, transitions between those states, and actions as well as for applications based on rules driven workflows.

A finite state machine consists of states, inputs, and transitions. A script in ACTA can describe a finite state machine as depicted in Figure 73, which illustrates a flowchart of a hero game. The flowchart consists of boxes for each action a hero can do such as standing, jumping, ducking, and diving. When acting, the hero can respond to a button press and perform a different action. Furthermore, in each of the depicted boxes, there are arrows connecting the current action with a next one according to the labeled pressed button. In detail, the ACTA script of this example consists of the following parts: a) a fixed set of states that the machine can be in (e.g., standing, jumping, ducking, and diving), b) current state, the machine can only be in one state at a time (a heroine can't be jumping and standing simultaneously), c) a sequence of inputs or events (e.g., the raw button presses and releases), and d) a set of transitions from current state to a new one based on incoming input or event. When an input or event comes in, if it matches a transition for the current state, the machine changes to the state that transition points to. An example of the hero game in ACTA script is depicted in Figure 74.

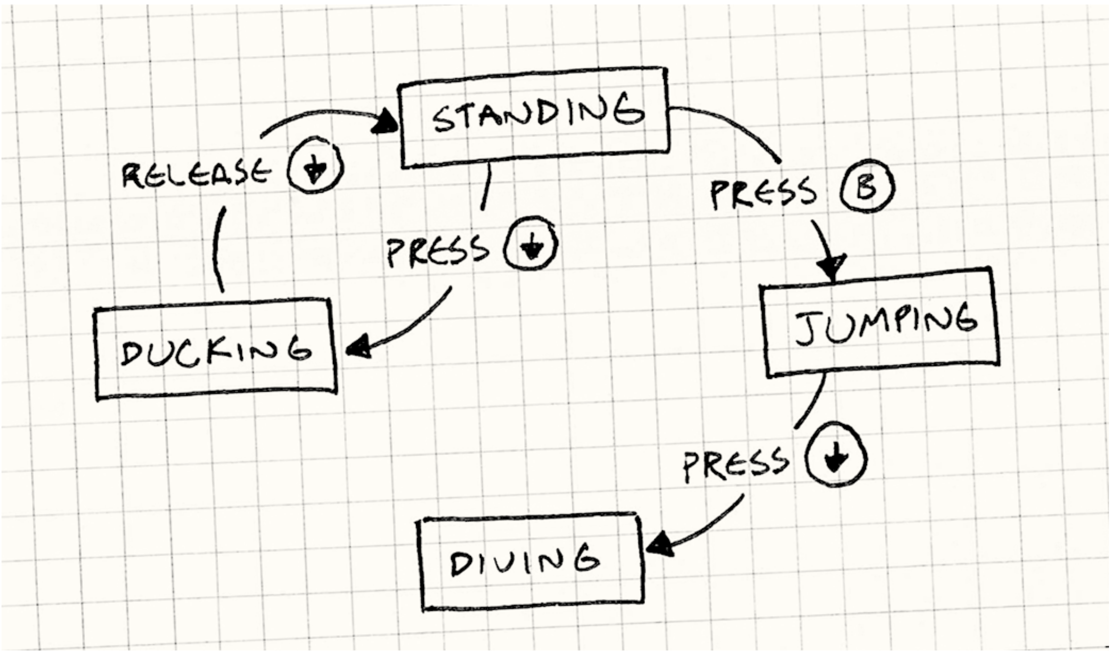


Figure 73: A Finite State Machine example[74]

```
State "STANDING"
{
    when (PressedButton == "ARROWDOWN") { NextState = "DUCKING"; }
    when (PressedButton == "B") { NextState = "JUMPING"; }
}

State "DUCKING"
{
    when (ReleasedButton == "ARROWDOWN") { NextState = "STANDING"; }
}

State "JUMPING"
{
    when (PressedButton == "ARROWDOWN") { NextState = "DIVING"; }
}
```

Figure 74: Finite states of an action game in ACTA script.

4.8.1 ACTA's syntax and semantics

Authors can use ACTA for scripting a smart game or application in general, by outlining a set of finite states and event-driven transitions (i.e., a transition from one state to another is triggered by an event or a message). Thus, ACTA facilitates event driven programming, in which the flow of the program is determined by events, event handlers, and asynchronous programming.

State	when	with	NextState
Name	Priority	Reevaluation	Active
true	True	false	False
null	Null	priority	name
active	Never	reevaluation	PreviousState
When_SetActive	When_SetReevaluation	When_SetPriority	

Table 14: ACTA keywords

[74] Image taken from http://www.padaonegames.com/bb/

As depicted in Figure 75, a script in ACTA initially consists of various transitions that may happen anytime during execution without caring about the current state (i.e., when statements). For example, when a user gets into the playing room, the game starts by switching the machine to the “init” state to initialize the game. Another example is that during play, when the user leaves the room, the machine goes to the “paused” state to handle this situation. In this state, the game gets paused until the user comes back (within a period of N seconds), returning the machine to the previous state. When the allowed time has elapsed, the game terminates by switching to the corresponding “end” state. The ACTA script continues with the definition of the states (i.e., State statements). To this end, ACTA uses 3 fundamental keywords for scripting a finite state machine: a) **State**, b) **NextState**, and c) **When**. Table 14 summarizes all keywords (or reserved words) used in ACTA script (keywords cannot be used as identifiers).

The **NextState** keyword is used to set a new state. Using an assignment expression, the author simply sets the id of the next state to the reserved word NextState (e.g., *NextState = “state id”*). Event-driven transitions are described using the keyword **when**. This keyword is used to define a when statement. In other words, when a condition becomes true, a transition from the current state to a new one is fired. As depicted in Figure 75, within the when statement’s block there are optional macros to be executed when exiting the state. To point to the new state, authors define a NextState statement at the end of the block.

A finite state is described using the keyword **State** that defines a state statement. As depicted in Figure 75, a state statement defines a block that contains: a) optional macros for execution when entering the state, and b) transitions. The latter may be either an explicit state change using keyword NextState (e.g., *NextState = “state’s id”*) or a set of when statements in order the machine to remain in the current state until the condition of a when statement becomes true and fires its transition.

Authors in ACTA are able to set the execution order of a when statement by setting its priority accordingly. **When** statements that have a higher priority value are executed first. The priority value can be positive or negative. The default value for any **when** statement is 0. Furthermore, authors are able to characterize a **when** statement with a unique id by setting its ***Name*** property. So, **when** statements that have the same priority are executed in the alphabetic order of their name properties.

Additionally, authors can set the re-evaluation behaviour of a **when** statement. In detail, they can specify whether a **when** statement can be re-evaluated by setting ***Always*** or ***Never*** to the **Revaluation** property (see section 4.8.2). **Revaluation** behaviour is used to limit the number of times that a **when** statement runs, primarily to prevent infinite loops (i.e., state A -> B -> C -> A etc.). ***Always*** (which is the default value) indicates that the **when** statement can be re-evaluated multiple times. ***Never*** indicates that the **when** statement is executed once. A condition may be evaluated several times until the **when** statement executes macros, but the **when** statement will never be evaluated again. Also, authors are able to disable a **when** statement by setting the value of its **Active** property to **True** (default) or **False. Active** property indicates whether the **when** statement should be evaluated. Setting to false this property is similar to comment out the **when** statement.

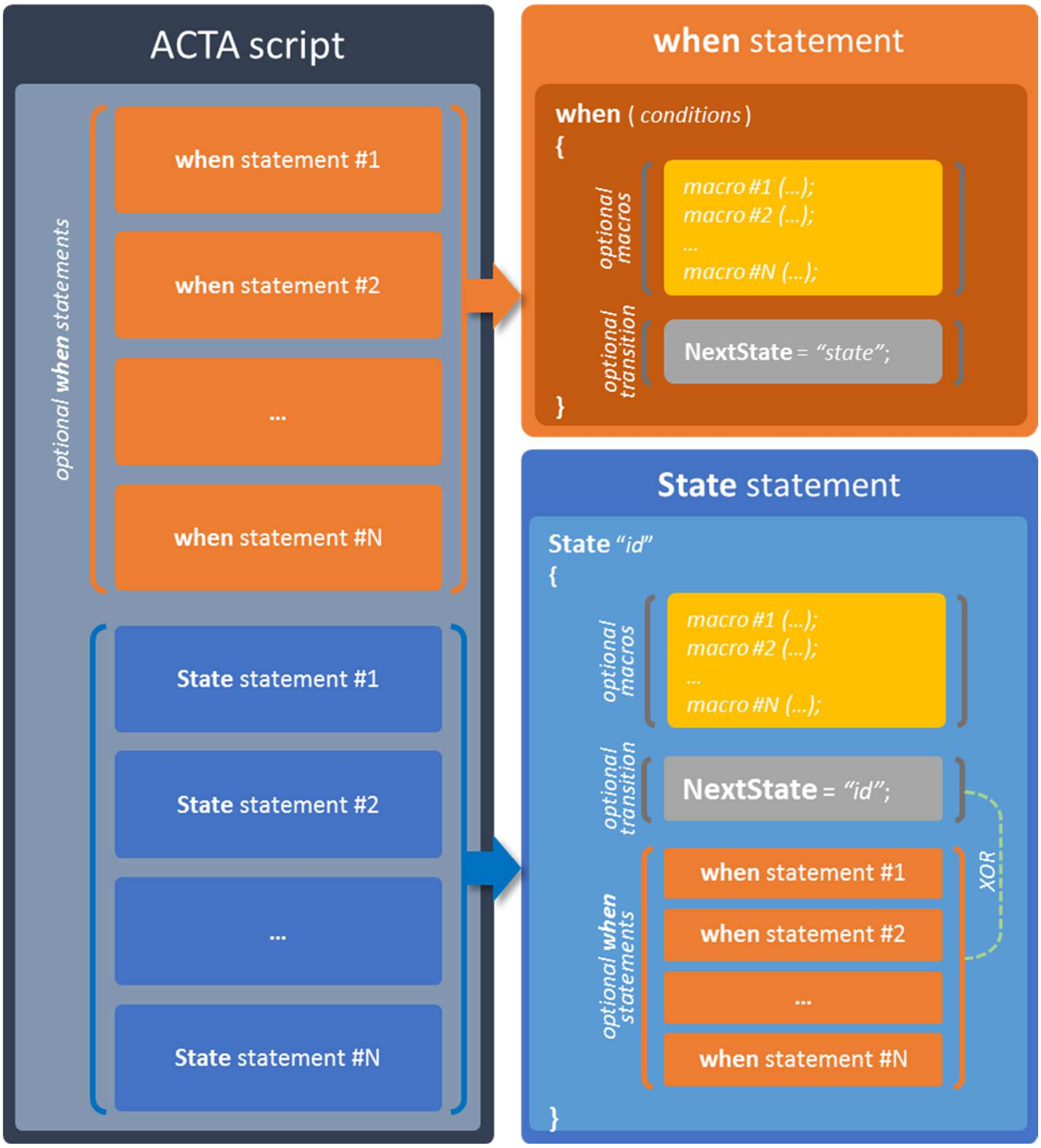


Figure 75: ACTA's structure

Finally, authors have access to a set of library functions to dynamically change the semantics of **when** statements. This set consists of the following library functions:

- **When_SetActive(bool *active*, params string[] *whenNames*):** sets the **Active** property of **when** statements with names presented into the list **whenNames** to the value given as a parameter (i.e., **active**).
- **When_SetReevaluation (string reevaluation, params string[] *whenNames*):** sets the **Revaluation** behaviour property of **when** statements with names presented into the list **whenNames** to the value given as a parameter (i.e., **reevaluation**).

- **When_SetPriority (int priority, params string[]** ***whenNames*****):** sets the **Priority** property (i.e. execution order) of **when** statements with names presented into the list **whenNames** to the value given as a parameter (i.e., **priority**).

Figure 76 illustrates the ACTA IDE which provides comprehensive facilities to authors for scripting in ACTA. The depicted example consists of two **when** statements of the same priority. Initially, the second one **when** statement (i.e., "when#2") is not active. When the value of the subtotal field becomes greater than 10000, when#1 executes its macros. The first one (i.e., *When_SetActive*) activates when#2, while the second one, sets the discount field equal to 0.5. So, when#1 reads the subtotal field and writes to the discount field, while when#2 reads the discount and subtotal fields and writes to the total field. Therefore, when#2 is active and has a dependency on when#1. As a result, when#2 is evaluated/reevaluated whenever when#1 executes its macros.

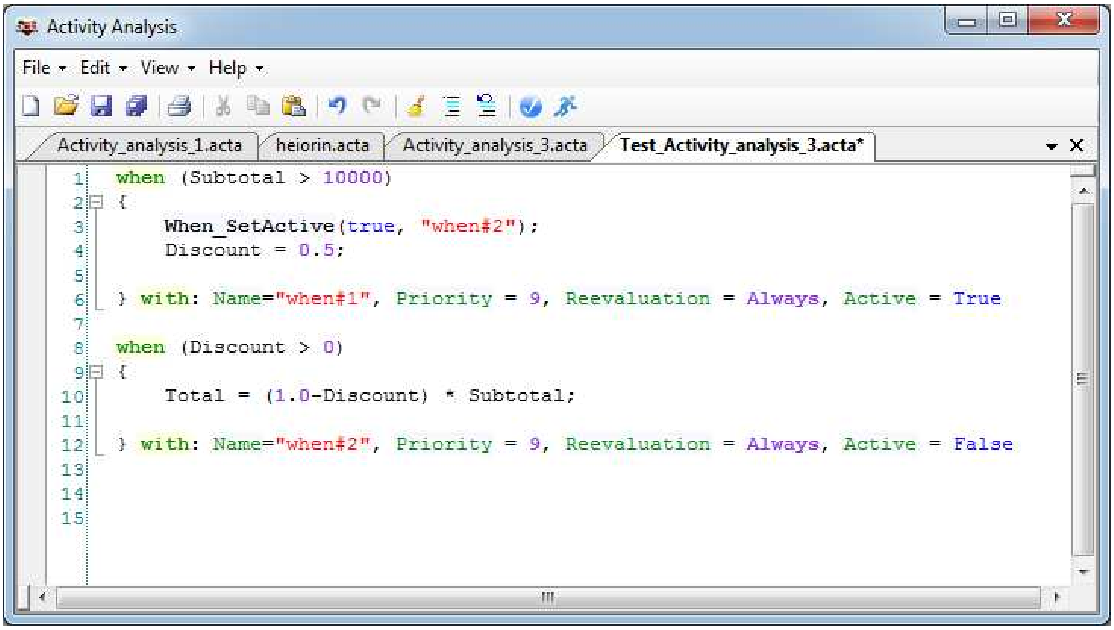


Figure 76: ACTA's semantics example

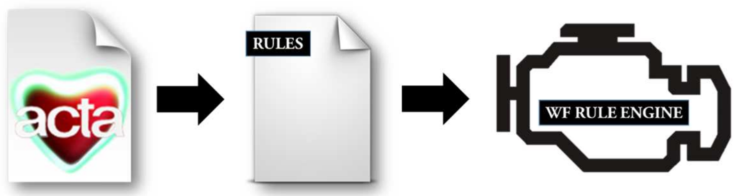


Figure 77: WF rule engine takes the ACTA generated ruleset for evaluation

4.8.2 ACTA from a developer's perspective

Windows Workflow Foundation[75] (WF) is the "glue layer" connecting ACTA's scripting vocabulary with the application internals. WF is a Microsoft technology that provides an in-process workflow engine to implement as workflows within .NET applications. Furthermore, WF provides rule capabilities to the .NET Framework development

[75] https://msdn.microsoft.com/en-us/library/vstudio/ms734631(v=vs.90).aspx

platform. These capabilities range from simple conditions that drive activity execution behavior to complex rulesets executed by a full-featured forward-chaining rules engine.

To this end, an ACTA script is converted to a WF ruleset in order to take advantage of the provided workflow rule engine supporting complex rules scenarios, demanding forward chaining evaluation and precise evaluation control. Figure 78 illustrates the ACTA IDE which provides comprehensive facilities to programmers for scripting in ACTA and generate the corresponding ruleset (filename extension *.rules*). A WF ruleset is a collection of rules with a set of execution semantics (e.g., If-Then-Else expressions that operate on properties within application). The generated rule expressions (i.e., ruleset) are given to the WF rule engine for evaluation as depicted in Figure 77.

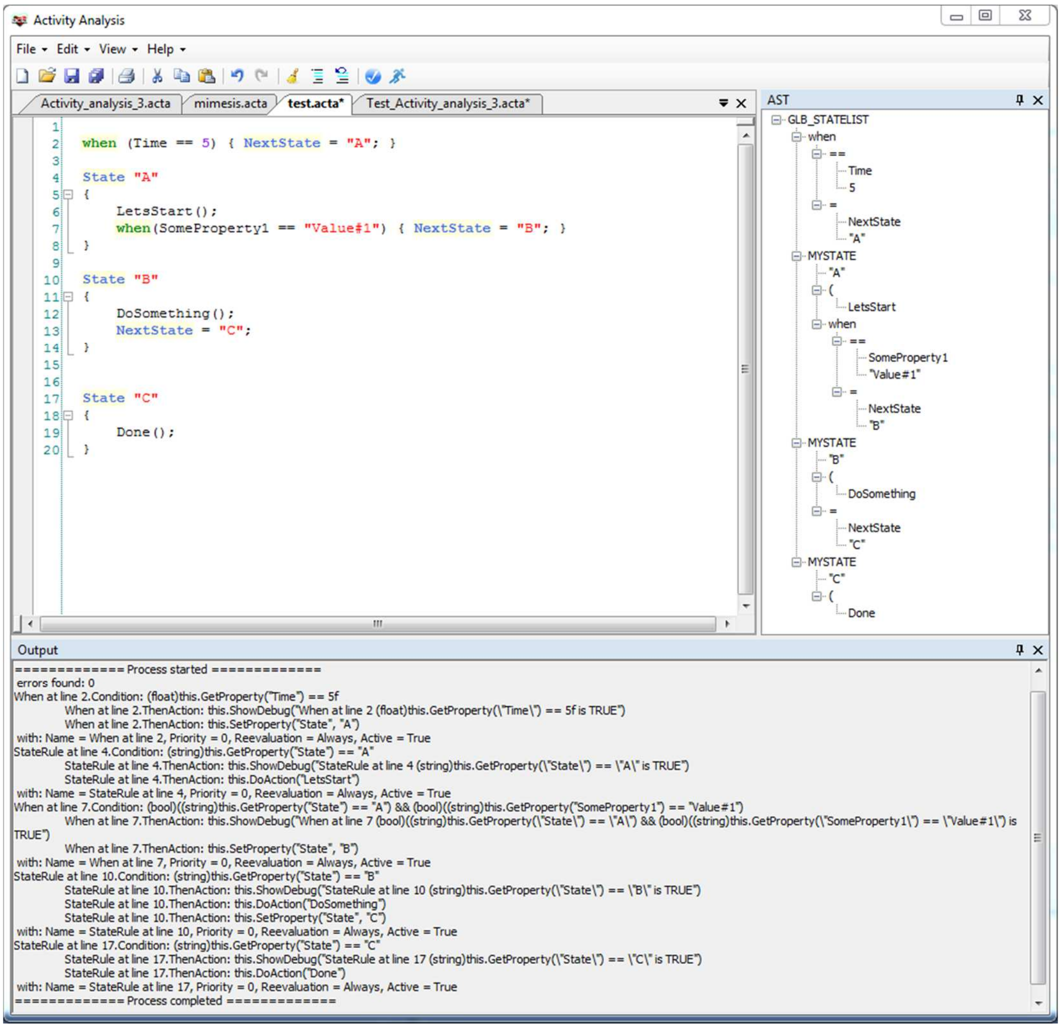


Figure 78: ACTA IDE

The ACTA IDE parses the statements found in an ACTA script and generates the equivalent ruleset through three distinct steps. Firstly, it parses ACTA script and translates each statement or expression found to its equal code expression using the Code Document Object Model (CodeDOM[76]). The .NET Framework includes this mechanism that enables developers to emit source code to generate source code in

[76] https://msdn.microsoft.com/en-us/library/650ax5cx(v=vs.110).aspx

multiple programming languages at run time, based on a single model that represents the code to render. For example, a logical expression in ACTA (e.g., x==5) is converted to an instance of CodeBinaryOperatorExpression with operator type of CodeBinaryOperatorType. ValueEquality (see CodeBlock 25). Secondly, a WF rule is created for each CodeDOM expression by defining a condition (i.e., r*ule expression condition*) with an associated set of actions to perform (i.e., *then actions*). Finally, the ACTA IDE creates a ruleset of the produced rules and uses WorkflowMarkupSerializer to write the ruleset to file. WorkflowMarkupSerializer provides methods that serialize workflow classes to XAML format.

In summary, ACTA is a general purpose finite state machine description language that facilitates the development of .NET applications by integrating into their projects the Windows Workflow Rule engine. In Windows Workflow Foundation all rulesets are defined in relation to a single specific .NET type. In case of ACTA scripting such a .NET type is not known in advance. To overcome this difficulty and allow dynamic linking between ACTA generated rules and multiple types, a reflection-oriented model was adopted and implemented as a .NET framework library presented in 4.9.2. To this end, variables used in ACTA are mapped to properties considered public to reflection (a property must have at least one accessor that is public). For example, the logical expression ***x==5***, part of the when statement (e.g., *when(**x==5**) { NextState = "A" }*) is equal to the following condition in CodeDOM: *(float)this.GetProperty("x") == 5f* (see CodeBlock 25).

```
<CodeBinaryOperatorExpression Operator="ValueEquality">
  <CodeBinaryOperatorExpression.Right>
    <CodePrimitiveExpression>
      <CodePrimitiveExpression.Value>
        <Single>5</Single>
      </CodePrimitiveExpression.Value>
    </CodePrimitiveExpression>
  </CodeBinaryOperatorExpression.Right>
  <CodeBinaryOperatorExpression.Left>
    <CodeCastExpression TargetType="System.Single">
      <CodeCastExpression.Expression>
        <CodeMethodInvokeExpression>
          <CodeMethodInvokeExpression.Parameters>
            <CodePrimitiveExpression>
              <CodePrimitiveExpression.Value>
                <String >x</String>
              </CodePrimitiveExpression.Value>
            </CodePrimitiveExpression>
          </CodeMethodInvokeExpression.Parameters>
          <CodeMethodInvokeExpression.Method>
            <CodeMethodReferenceExpression MethodName="GetProperty">
              <CodeMethodReferenceExpression.TargetObject>
                <CodeThisReferenceExpression />
              </CodeMethodReferenceExpression.TargetObject>
            </CodeMethodReferenceExpression>
          </CodeMethodInvokeExpression.Method>
        </CodeMethodInvokeExpression>
      </CodeCastExpression.Expression>
    </CodeCastExpression>
  </CodeBinaryOperatorExpression.Left>
</CodeBinaryOperatorExpression>
```

CodeBlock 25: Rule condition expression using CodeDOM

4.8.3 Implementation details

The ACTA IDE was implemented in .NET using C# and uses a parser generated with ANTLR[77] to read and process ACTA scripts. ANTLR (ANother Tool for Language Recognition) is a powerful parser generator for reading, processing, executing and translating structured text or binary files. It is widely used to build languages, tools, and frameworks. From a grammar, ANTLR generates a parser that can build and walk parse trees. Taking as input the ACTA grammar (see example in CodeBlock 26), ANTLR generates a lexer and a parser for that language that can automatically build parse-trees, which are data structures representing how a grammar matches the input. ANTLR also automatically generates tree walkers that can be used to visit the nodes of those trees to execute application-specific code (see APPENDIX I – ACTA grammar).

```
//Rules
program : whenstmt* mystate* EOF -> ^( GLB_STATELIST whenstmt* mystate*);
mystate : 'State' STRING '{' stmt2* whenstmt* '}' -> ^(MYSTATE STRING stmt2*
whenstmt*)
        ;

whenstmt : WHEN '(' logicalExpression ')' '{' stmt2* '}' whenstmtattr? ->
^(WHEN logicalExpression stmt2* whenstmtattr?)
        ;

whenstmtattr: 'with' ':' whenid ',' priority ',' reevaluation (','
whenactive)?-> ^(WHENSTMTATTR whenid priority reevaluation whenactive?);

whenid : ('Name' | 'name')! ASS! STRING;

priority : ('Priority' | 'priority')! ASS! NUM;

reevaluation: ('Reevaluation' | 'reevaluation')! ASS! reevaluationAttr;

reevaluationAttr: 'Always' | 'Never';

whenactive: ('Active' | 'active')! ASS! BOOLEAN;
```

CodeBlock 26: An example of ACTA grammar

The ACTA IDE provides authors with SyntaxBox[78]; a syntax highlight editor. SyntaxBox features powerful syntax highlight windows forms control for the Microsoft.NET Platform. It supports: a) syntax highlighting, b) code folding for just about any programming language, c) bracket matching, d) unlimited undo/redo buffer, e) bookmarks, etc.

4.9 Bean SDK: A developer's toolkit

A software developer toolkit (Bean SDK) that allows the creation of smart games was developed for the proposed framework. Bean SDK consists of application programming interfaces (APIs) in the form of libraries implemented with .NET framework using C# programming language. Additionally, it provides sophisticated software modules to facilitate the communication and interoperability with the implemented augmented

[77] http://www.antlr.org/
[78] https://code.google.com/p/alsing/wiki/SyntaxBox

artifacts and services. Finally, Bean SDK located in the assembly BigBean.dll, provides developers with tools and utilities to integrate workflow activities into their projects.

4.9.1 ServiceManager

ServiceManager (namespace Bean.Core.ServiceManager) facilitates the operation, administration, maintenance and provisioning of middleware services implemented for the purposes of the current thesis. In detail, **ServiceManager** is a singleton class that keeps internally a service directory with service entries. Each entry has information about the following: a) running context, b) service object reference, c) event handler object, and d) interface type.

```
public bool EnableService<T>(string context, object handlerObj) where T: class
public bool EnableService<T>(string context) where T : class
public bool DisposeService<T>(string context) where T : class

public T Service<T>(string context)
public T2 EventHandlerForService<T1, T2>(string context)
```

CodeBlock 27: ServiceManager API

As depicted in CodeBlock 27, ServiceManager provides developers with the following functionality:

- **EnableService:** adds a new service entry to the service directory and resolves the service of the specified type which runs in context given as a parameter. The process returns *FALSE*, if the service entry already exists or if it has failed to resolve service. Otherwise, the process returns *TRUE*. The overload version of **EnableService** associates the requested service with the event handler object given as a parameter for later reference.
- **DisposeService:** removes the new service entry from the service directory and disposed resolved service which runs in context given as a parameter. The process returns *FALSE*, if the service entry was not found or if it has failed to dispose the service. Otherwise, the process returns *TRUE*.
- **Service:** lookup the service directory until the corresponding service entry is found and returns the service object reference of the specified type which runs in context given as a parameter. If the lookup procedure fails, the function returns *NULL*.
- **EventHandlerForService:** lookup the service directory until the corresponding service entry is found and returns the event handler object for the service of the specified type which runs in context given as a parameter. If the lookup procedure fails, the function returns *NULL*.

```
Bean.Core.ServiceManager.Instance.EnableService<Beantable.Max>("bean");
Bean.Core.ServiceManager.Instance.Service<Beantable.Max>("bean")
                                             .ShowMedia("intro.mp4", out e);
Bean.Core.ServiceManager.Instance.DisposeService<Beantable.Max>(" bean ");
```

CodeBlock 28: Using ServiceManager; a code sample

For example, a smart game developer decides to employ Max (child's virtual partner) in order to present multimedia content to the television next to him. As CodeBlock 28 illustrates, the developers proceed with the following distinct three steps: a) enable Max middleware service, b) use service to call the remote procedure *ShowMedia*, and c) dispose Max middleware when no longer needed.

4.9.2 WorkflowSystem

WorkflowSystem (namespace Bean.Core.WorkflowSystem) is a C# class library that integrates: a) a Workflow Foundation rules engine for the evaluation of ruleset produced by ACTA script, b) the implementation of ACTA's library functions (see CodeBlock 29), and c) a reflection-oriented model mechanism for dynamic loading and usage of assemblies which their type declaration in not known in ACTA script. The latter allows dynamic linking between ACTA generated rules and properties and methods found in assemblies considered public to reflection types through the rule evaluation process.

```
public void When_SetPriority(int priority, params string[] ruleNames)
public void When_SetReevaluation(string reevaluation, params string[]
ruleNames)
protected void When_SetActive(bool active, params string[] ruleNames)
```

CodeBlock 29: WorkflowSystem implements ACTA's library functions

```
public void LoadRulesFromFile(string ruleSetName, string rulesFilePath)

public void Start()
public void Stop()

public void Register<T>(T module)
public void UnRegister<T>(T module)
```

CodeBlock 30: WorkflowSystem rules engine functionality and reflection-oriented programming interface

To begin with, **WorkflowSystem** provides developers with functionality to import a file with rules exported from ACTA IDE (filename extension is *.rules*) or to start/stop the integrated rules engine. Additionally, it allows developers to register or unregister modules which are instances of classes (found in assemblies) that contain properties and methods used in a certain ACTA script. Prerequisite for each module is to implement the interface INotifyPropertyChanged. The latter notifies clients that a property value has changed. To that end, each module can either implement that interface or derive from **WorkflowSystem** provided wrapper called INotifyPropertyChangedWrapper. The latter is a generic property-change notification mechanism which implements the aforementioned interface and raises a PropertyChanged event when a property is changed. **WorkflowSystem** adds in a built in event-queue incoming property-change notifications for processing and rules evaluation.

As depicted in CodeBlock 30, **WorkflowSystem** provides developers with the following functionality:

- **LoadRulesFromFile:** loads the serialized ruleset from the file location given as a parameter. The WorkflowMarkupSerializer class is used to deserialize a ruleset exported into XML format by ACTA IDE.
- **Start:** creates and starts a thread which continually checks for property-change notifications in the event-queue and triggers the rules engine to evaluate the active rule instances.
- **Stop:** suspends the running thread created by the function **Start,** which has as a result to stop the rules engine evaluation process.
- **Register:** observes (using Reflection[79]) the structure and the behaviour of the (i.e. class instance) given module as a parameter and adds it in a built-in list for later method invocations. Furthermore, it subscribes to module's property-change notifications by adding an event handler to the module's PropertyChanged event.
- **UnRegister:** unsubscribes from the module's property-change notifications and remove the module given as a parameter from the internal list.

WorkflowSystem facilitates the connection between variables and macros used in ACTA with properties and methods found in the registered modules. As depicted in CodeBlock 31, the connection is facilitated by the following functionality which employs C# Reflection techniques:

- **DoAction:** propagates an ACTA macro to its corresponding method invocation. In detail, it searches the built-in list of registered modules to find the module which has a public method to reflection with name given as a parameter. Afterwards, it proceeds with method invocation passing the given arguments.
- **GetProperty:** searches the built-in list of registered modules to find the module which has a property with name given as a parameter and with at least one public getter accessor. If found, it proceeds with get property invocation to get its current value.
- **SetProperty:** searches the built-in list of registered modules to find the module which has a property with name given as a parameter and with at least one public setter accessor. If found, it proceeds with set property invocation to set its current value with this given as a parameter.

For example, an ACTA logical expression *x==5,* results to the function call of GetProperty to set the value of property x defined in a registered module equal to 5. (i.e., *(float)GetProperty("x") == 5f)*. Similarly, the assignment expression *x=5* is equal to*SetProperty("x", 5f)* and the macro in ACTA, *print("hello")* is equivalent to *DoAction("print", "hello");*

```
protected Object DoAction(String name, params object[] args)
protected object GetProperty(String name)
protected void SetProperty(String name, object value)
```

CodeBlock 31: WorkflowSystem reflection based mechanism for connection between ACTA script and registered modules

[79] https://msdn.microsoft.com/en-us/library/ms173183.aspx

Finally, **WorkflowSystem** models a finite state machine by introducing the class **StateManager**. The general purposes of ACTA script are served with properties defined within this class such as **State** or **PreviousState**. To this end, **StateManager** provides a finite state machine scripted in ACTA with information about: a) the current state, b) the previous one state (i.e., property PreviousState), and c) the count of transitions to the current state (i.e., property Count). Additionally, the property **Time** measures elapsed time in seconds which is reset after performing a state transition. As CodeBlock 32 depicts, **StateManager** is automatically registered to **WorkflowSystem** as module which derives from INotifyPropertyChangedWrapper.

```
public class StateManager : INotifyPropertyChangedWrapper
{
     public int Time
     public int Count

     public String PreviousState
     public String State
}
```

CodeBlock 32: ACTA's state manager provided with WorkflowSystem

4.9.3 Modules

Bean SDK is provided with a large set of modules facilitating the easy integration in smart games of low level communication with middleware services, as well as complex higher level functionality and interoperability. Each module is derived from INotifyPropertyChangedWrapper and therefore is ready for registration in WorkflowSystem and subsequently to be used in ACTA scripts. The modules provided with Bean SDK include:

- **SmartChairForcePressureRecognition:** beyond basic functionality exposed by the corresponding middleware service (as presented in 4.5.4), this module detects and helps authors to describe in ACTA when a UI component is hit by the point provided as the center from the force-pressure distribution information (e.g., *IsSmartChairHit(buttonOK) == true*).
- **FingerRecognition:** beyond basic functionality exposed by the physical object and cursor recognition service (as presented in 4.5.1.3), it helps authors to describe in ACTA when a user touches a UI component (e.g., *OnTouchDown(buttonOK) == true*).
- **GestureRecognition:** beyond basic functionality exposed by the corresponding Nibbler infrastructure's sensory module (as presented in 4.4.1.2), it uses gestures and recognized body postures to extract complex information about body language. Therefore, authors are able to describe in ACTA when a user is attentive and is interested in the interaction (e.g., *UserIsInterested == false*).
- **Mate3D:** is a wrapper of the virtual character's exposed functionality as presented in 4.7.2, enriched with capabilities for text to speech (TTS) operations such as preferred voice settings and dynamic language selection. Additionally, the wrapper offers asynchronous text to speech functions which store TTS data in a queue and upon speech completion, the next sentence is automatically sent for reading. For

example, the two ACTA expressions *Lang="GR" and SpeakAsync("leavingQuestion")* will have as a result Max to ask in the Greek language if the child is leaving the room.

- **PhysicalObjectRecognition:** beyond basic functionality exposed by the physical object and cursor recognition service (as presented in 4.5.1.3), it extracts information about the position of an object in relation to other physical objects. In detail, it is based on an XML file (created with an authoring tool for tangible games presented later in 4.10.1) that defines physical objects' identification numbers, relevant positions and the relationship among them. Therefore, it helps authors to describe in ACTA when the user matches a physical object with its adjacent pieces. For example, *IsPhysicalObjectWithIDMatchedWithAdjacentPieces(3) == true*.
- **SessionManager:** beyond basic functionality exposed by the adaptation infrastructure mechanism ADAM (as presented in 4.3.1), it provides supplementary functions for pausing and resuming the current session.
- **SmartPen:** beyond basic functionality exposed by the corresponding middleware service (as presented in 4.5.3), it records the pressure during writing and extracts indications. Therefore, the SmartPen module helps authors to describe in ACTA when the child's applied pressure on the pen's tip was not stable during the drawing of a line (e.g., *LastDrawingPressureNotChangedOverThreshold(2) == true;*).
- **SpeechRecognition:** beyond basic functionality exposed by the corresponding Nibbler sensory infrastructure's sensory module (as presented in 4.4.1.4), it performs statistical analysis of how often the user talks and recognizes if he is a very active in talking or he has not spoken at all. For example, *UserDoesNotTalk == true*.
- **SurfaceForcePressureRecognition:** beyond basic functionality exposed by the corresponding middleware service (as presented in 4.5.4), it detects and helps authors to describe in ACTA when a UI component is hit by the point provided as the center from the force-pressure distribution information (e.g., *IsSmartChairHit(buttonOK) == true*).
- **UserPosition:** beyond basic functionality exposed by the corresponding Nibbler infrastructure's sensory module (as presented in 4.4.1.4), it identifies when the user is presented within the interaction area or getting very close to the employed Kinect sensor. Therefore, UserPosition module helps authors to describe in ACTA when the child is leaving or entering the interaction area, e.g., *when(!UserPresent)* or *when(UserTooFar).*

4.10 Administration facilities

The proposed framework employs a large set of administration facilities, such as monitoring and authoring tools, as well as content administration facilities. These tools are intended for use by developers, administrators, therapists and parents. Game developers can use an authoring tool for the creation of tangible games. Additionally, they can get an overview of the sensing capabilities the Beantable's embedded sensory infrastructure. System administrators (i.e., technical support) are supported through a tool which runs in the background and monitors the employed services or applications supposed to stay running. Early intervention professionals and parents are provided with an administration suite for content editing and monitoring of child's development progress (i.e., play maturity), which fits the interests of both.

4.10.1 Authoring tool for tangible games

An authoring tool was created to facilitate the creation and administration of games based on physical object matching like jigsaw puzzles or matching cards. The tool receives fiducial markers on the physical objects' abutting side and creates an XML file (filename extension is *.bean*) that defines their identification numbers, relevant positions and the relationship among them (see Figure 79).

```xml
<?xml version="1.0"?>
<Game>
  <PhysicalObjects>
    <PhysicalObject id="26">
      <point x="0,230" y="0,341" xOffset="0,008" yOffset="-0,009" />
      <orientation angle="1,553" angleOffset="0,061" />
      <image path="3pigs_pieces_150dpi__0001_Layer 2.png" dpi="244" />
      <origin x="0,427" y="0,547" />
    </PhysicalObject>
    <PhysicalObject id="28">
      <point x="0,458" y="0,339" xOffset="-0,002" yOffset="0,011" />
      <orientation angle="1,584" angleOffset="0" />
      <image path="3pigs_pieces_150dpi__0003_Layer 4.png" dpi="244" />
      <origin x="0,429" y="0,511" />
    </PhysicalObject>
    <PhysicalObject id="34">
      <point x="0,334" y="0,315" xOffset="0,014" yOffset="0,047" />
      <orientation angle="2,115" angleOffset="-0,563" />
      <image path="3pigs_pieces_150dpi__0010_Layer 13.png" dpi="244" />
      <origin x="0,248" y="0,561" />
    </PhysicalObject>
    <PhysicalObject id="35">
      <point x="0,319" y="0,50" xOffset="-0,019" yOffset="0,001" />
      <orientation angle="5,578" angleOffset="-2,383" />
      <image path="3pigs_pieces_150dpi__0009_Layer 14.png" dpi="244" />
      <origin x="0,616" y="0,479" />
    </PhysicalObject>
  </PhysicalObjects>
  <Neighborhood>
    <Neighbors A="26" B="34" />
    <Neighbors A="34" B="28" />
    <Neighbors A="34" B="35" />
  </Neighborhood>
</Game>
```

Figure 79: XML file that models physical objects and their relationships.

The creation of the configuration file is completed in four simple guided steps. In the first step, the author places each physical object on the Beantable's surface and selects a representative image (see Figure 80 A). Object's alignment may be needed in case the image size does not match to the size of the actual physical object or in case the pivot point of the fiducial marker is different from the image's pivot point which defaults to its geometrical center (see Figure 80 B).

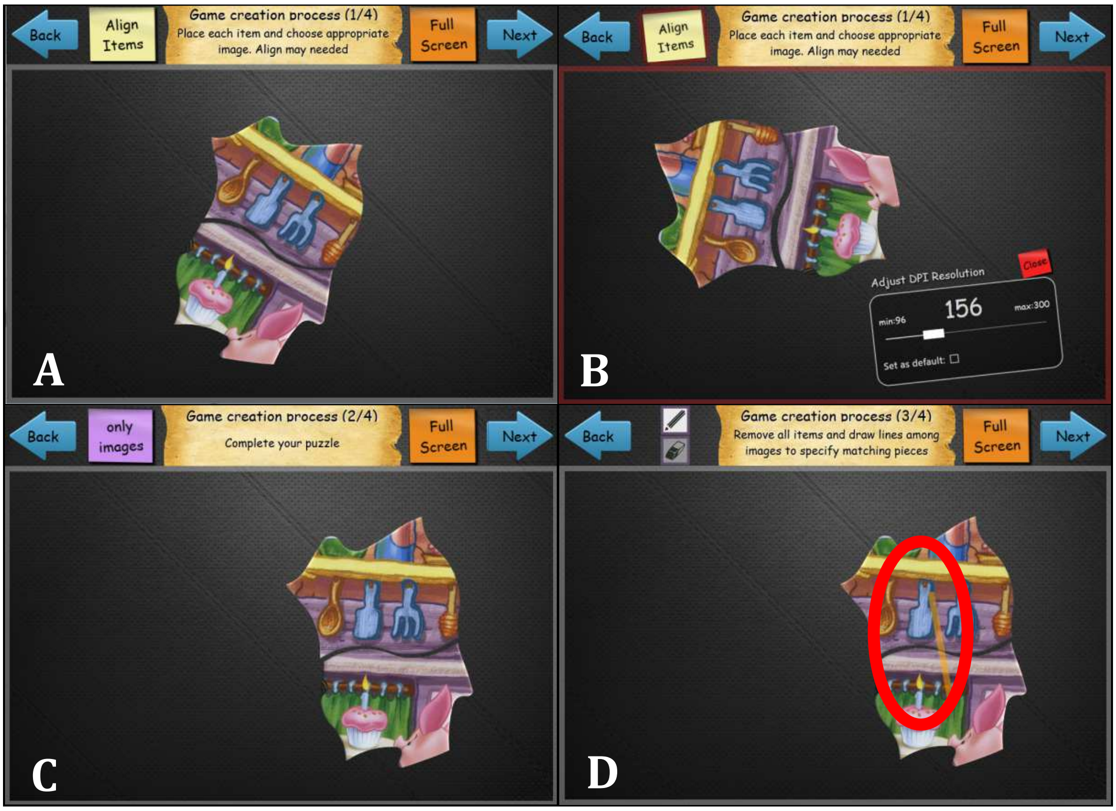


Figure 80: Authoring tool for tangible games – Part A

Figure 81: Authoring tool for tangible games – Part B

In the second step, the author matches the adjacent pieces with each other by physical object manipulation (see Figure 80 C). In the following step, the author removes the physical objects and uses the tool to mark the matching pairs (see Figure 80 D). In the last step, the author has the option to save the configuration file (see Figure 81 E).

Finally, the authoring tool provides the option to load and test the effectiveness of the calibration file of any saved tangible game (see Figure 81 F). Extra visual cues are provided by the administration software to help the authors visualize the matching pair of every piece. For example, when the author rotates a physical item around itself, the outline of its adjacent pieces is illuminated in pink (see Figure 81 G). Similarly, when the author matches successfully two adjacent pieces, their outline is illuminated in green as depicted in Figure 81 (H).

4.10.2 Sensors monitoring

Beantable is accompanied by the **Sensors monitoring** utility that can be used for monitoring the captured data of the installed hardware sensors. The utility provides an overview of the captured data received from each available service such as: a) physical object and cursor recognition service (as presented in 4.5.1.3.2), b) augmented pen (see 4.5.3), c) augmented chair (see 4.5.4), and d) force pressure recognition service (see 4.5.1.4).

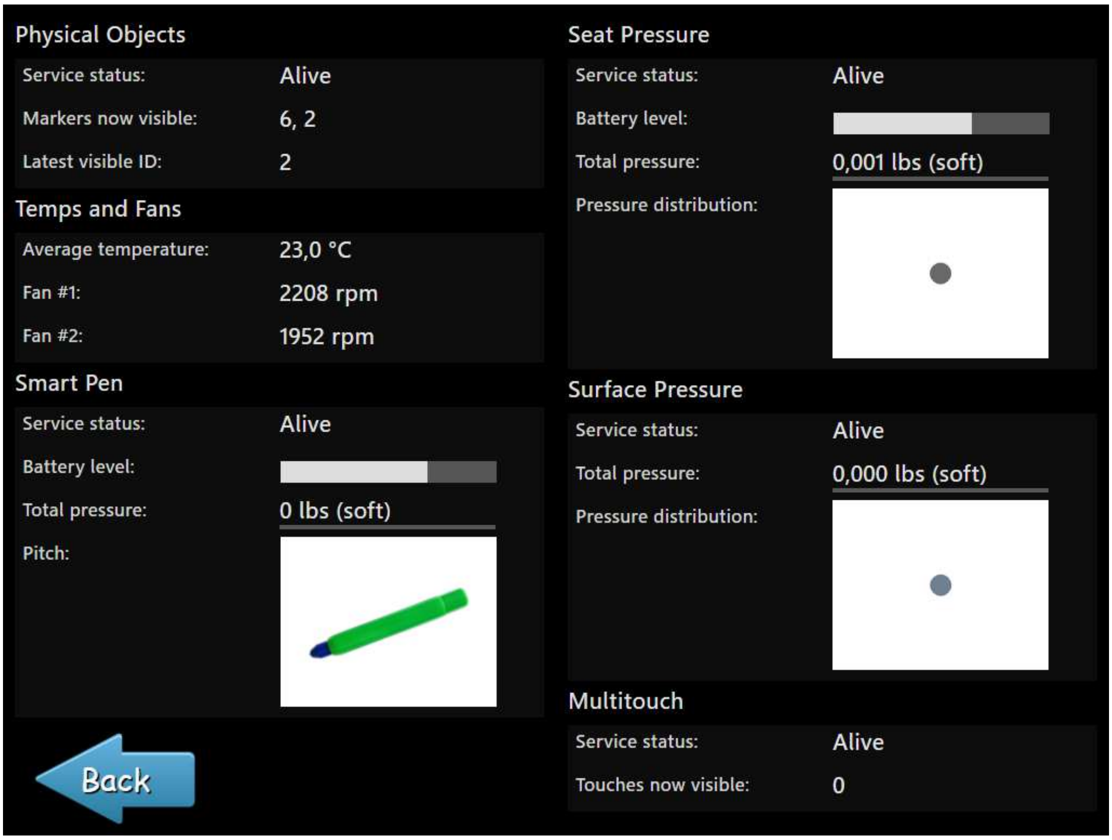


Figure 82: Sensors monitoring utility

The **Sensors monitoring** utility displays information about each service status (*Alive* or *Dead*). As depicted in Figure 82, all the aforementioned services are up and running (e.g., *Alive*). In detail, Figure 82 depicts that two physical objects are currently placed on the Beantable's surface of which the last placed has a fiducial marker of id 2 on its abutting side. Furthermore, the operating temperature inside Beantable is about 23,0 °C while the speed of the install cooling fans is 2208 and 1952 rpm respectively. Regarding the augmented pen, its battery level is more than half while the pen is almost in a horizontal position and its tip is not pressed (i.e., 0 lbs). In the right half of the figure there is a preview for the distribution info captured by both the Beantable's surface and the augmented chair. As illustrated in the right upper half, the battery level of the XBee module embedded under the chair is more the half, while it is also seen that nobody sits interpreting the force pressure distribution which is depicted just in the center of the white panel (the white panel simulates the chair's sitting area). Similarly, the same information goes for the Beantable's surface captured pressure. Finally, as depicted in the bottom half, the counter of touches detected by the vision sub-systems is equal to 0 meaning that nobody is currently touching the Beantable's surface.

4.10.3 Click & Go

In many cases it can be a source of frustration if the software running in a computer crashes or hangs just after users leave it unattended. The proposed framework delivers a tool, called Click & Go, which is responsible to keep the employed services or applications up and running. The middleware services and applications need to keep running at all times. To this end, Click & Go offers a feature of monitoring and re-starting them whenever they are closed or they have crashed.

Click & Go is a simple tool that automatically restarts selected processes when it detects that they have stopped. It is useful for Beantable's background programs that occasionally crashes. Ensuring that critical processes do not become unavailable can also be accomplished with this tool. That way computer vision sub-systems and middleware services will keep restarting to continue their tasks or until users stop Click & Go manually.

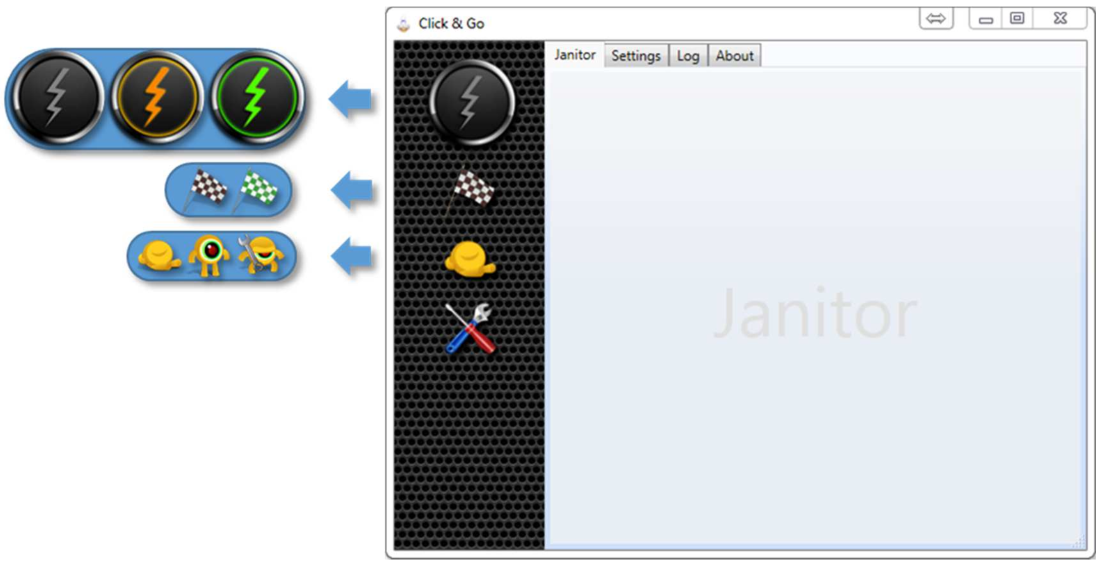


Figure 83: Click & Go

Click & Go uses the term "job" to describe either a standalone application (e.g., computer vision sub-systems) or an application which runs as a middleware service (e.g., physical object and cursor recognition). Additionally, the same term is used to describe a middleware service dependency meaning that the corresponding job is not intended to run on the local machine but somewhere else on the network. As Figure 83 illustrates, Click & Go supports three modes of monitoring (i.e., janitor mode): a) disabled, b) monitoring, and c) rectification. Click & Go uses various color codes to preview a job status:

- **green**: means that Click & Go started a job and its process is currently running.
- **yellow**: means that Click & Go found that middleware service is running somewhere else on the network.
- **red**: means that Click & Go found that the job is not running or the middleware service dependency is not available (i.e., not running somewhere else on the network).
- **grey**: the job is disabled.

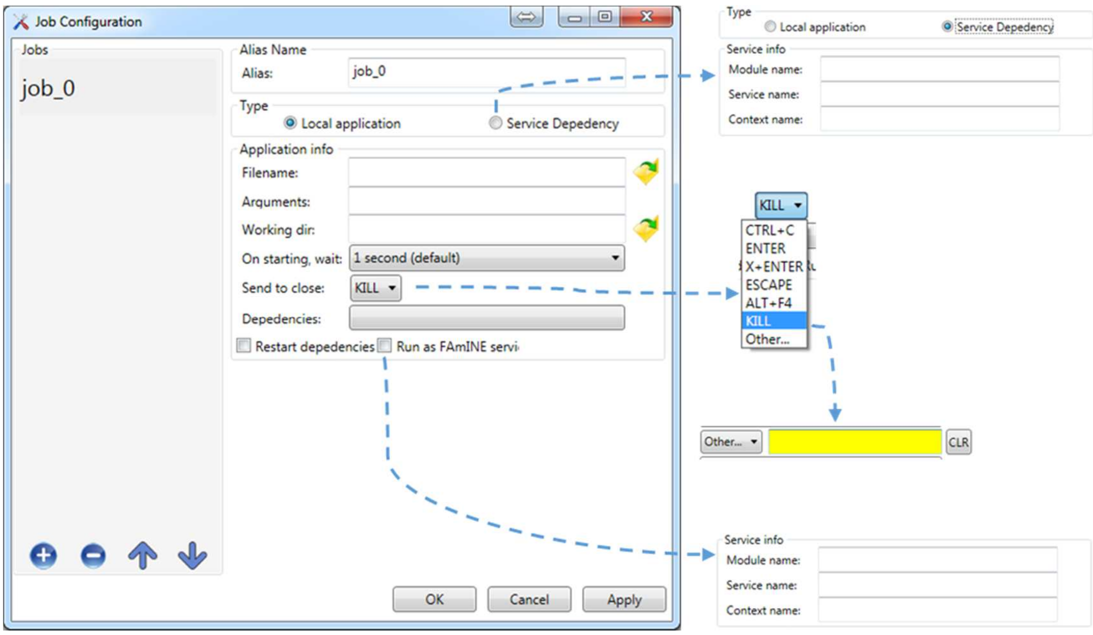


Figure 84: Click & Go Job Configuration

Authors are able to add a new job by using the job configuration tool as depicted in Figure 84. In detail, they choose the type of the job, i.e., local application or middleware service dependency. Authors can add further details such as running context and interface name in case of a local application which exposes functionality to the network (i.e., as a middleware service). Furthermore, authors set local application details such as: a) executable, b) arguments, c) working directory, d) waiting time before janitor starts monitoring, and e) job dependencies which is a list of other jobs that they have to be up and running. Authors can set Click & Go to restart job dependencies in case of job crashes, as well as keys combination for closing when needs. The drop down list of keys combinations includes: a) CTRL+C, b) ALT+F4, c) ESCAPE, d) ENTER, e) X+ENTER or f) other custom keys combination that can be recorded directly.

Click & Go runs as a middleware service by itself for handling requests from remote clients about starting or stopping defined jobs. Additionally, when it starts a service dependency service, it requests other Click & Go instances on the network to start their local application which is able to run as a middleware service. To this end, Click & Go orchestrates middleware services available in an AmI environment with respect to their dependencies. For example, Click & Go running on machine "A" tries to start a software program "SA", which depends on the physical object and cursor recognition service. Unfortunately, there is no job defined on machine "A" that is able to satisfy this service dependency. Thus, Click & Go asks remotely other instances if they are able to start the required service. Click & Go on machine "B" receives the request and before starting the physical object and cursor recognition service, it starts its service dependencies (as presented in 4.5.1.3.1) such as: a) **CvUtils**, b) **VirtualCamera**, c) **ImageNormalizer**, and d) **reacTIVision**. Subsequently, Click & Go on machine B starts the service and Click & Go on machine A continues and starts software program "SA".

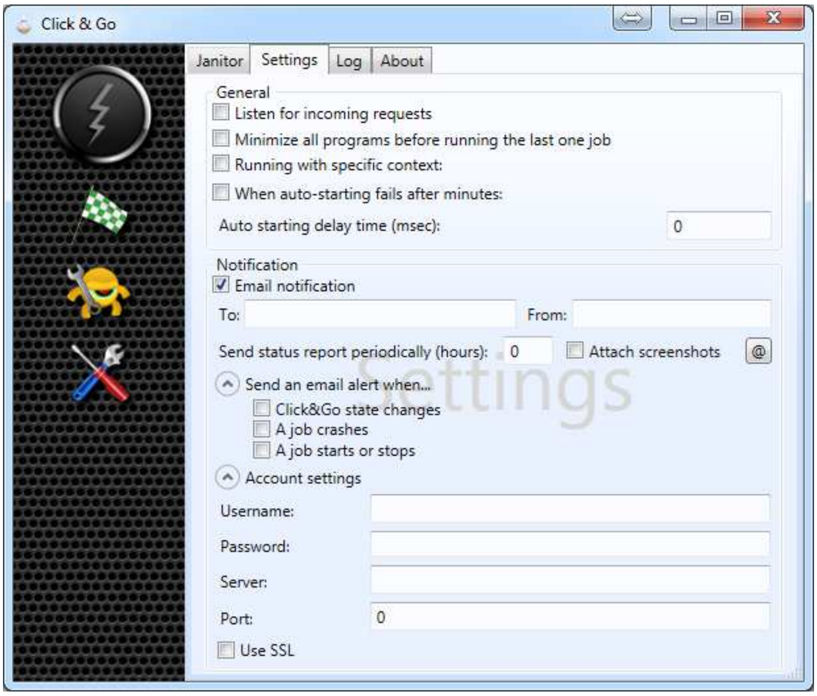


Figure 85: Click & Go Settings

As depicted in Figure 85, Click & Go keeps history logs about jobs' launches and crashes and can send notification emails to the technical support. Additionally, authors can set Click & Go to not listening on incoming requests or waiting a period of seconds before starting defined jobs. For example, when Click & Go starts simultaneously on computer booting (green flag), it may need some time for network initializing.

4.10.4 Bean Administration Suite

The proposed framework is delivered with a software suite called Bean Administration Suite. This suite consists of content administration facilities and monitoring utilities suitable for use by early intervention professionals and parents. The suite was implemented in .NET framework using C# programming language, and employs the

Microsoft Entity framework to access the existing database (i.e., the **Bean model)**. Entity Framework (EF) is an object-relational mapper that enables .NET developers to work with relational data using domain-specific objects. The Bean Administration Suite is adapted to the different needs of the users after they login (see Figure 86).

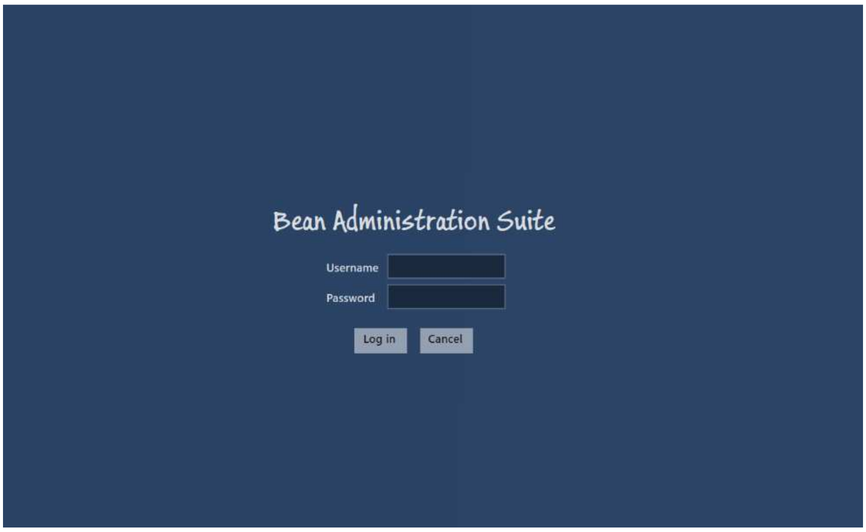


Figure 86: Bean Administration Suite (login screen).

4.10.4.1 Bean editor

Early intervention professionals and system administrators can use Bean Administration Suite for content editing of a) ICF-CY activities and body functions targeted by the proposed framework, b) body functions involved in an activity, c) smart games regarding their description and the activities they require, d) child's profile including limitations in activities and body structure problems, and e) human and system tasks.

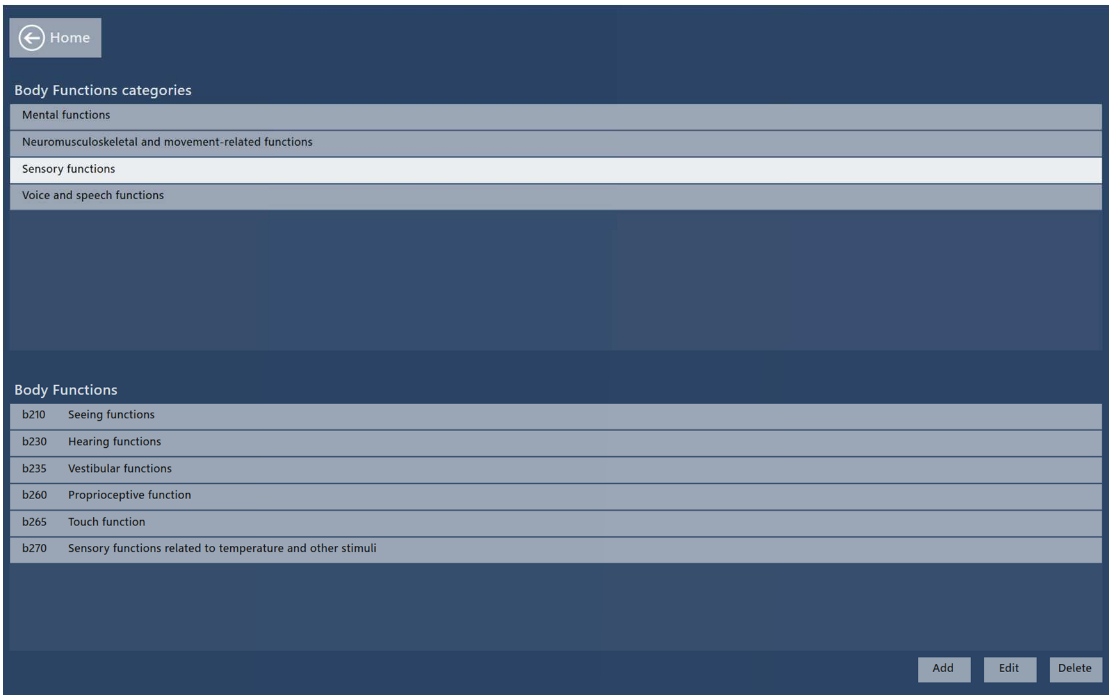


Figure 87: Bean Administration Suite (ICF-CY body functions).

As depicted in Figure 87, authors are able to view, edit or delete an ICF-CY body function of a selected category. In detail, authors can set the name, the code, and the description of a body function according to the ICF-CY (see Figure 88).

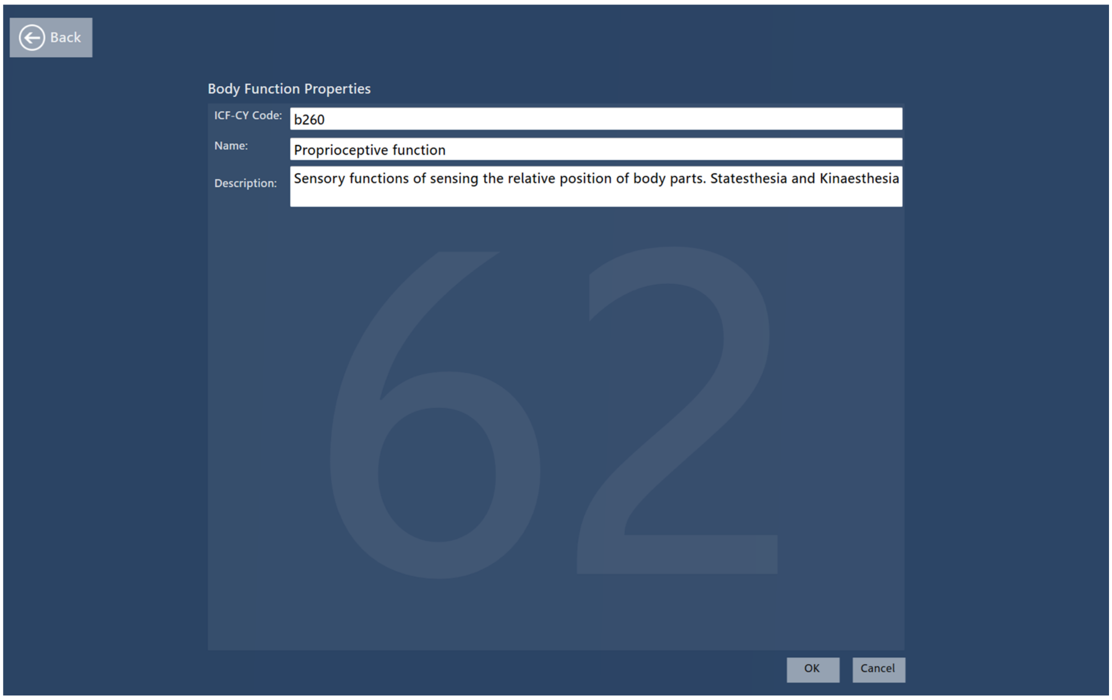


Figure 88: Bean Administration Suite (ICF-CY body function properties).

Figure 89 Bean Administration Suite (human tasks)

Regarding human and system tasks, authors are able to add, edit or remove a task which is characterized simply by a textual description (see Figure 89and Figure 90).

Figure 90: Bean Administration Suite (system tasks)

Similarly, authors are able to view, edit or delete an ICF-CY activity of a selected category as depicted in Figure 91. In detail, authors can set the name, the code, and the description of an activity according to the ICF-CY (see Figure 92). Additionally, they can set the expected capacity by specific age based on the Denver's II scale. Moreover, authors are able to select among the relevant human and system tasks, as well as the involved body functions.

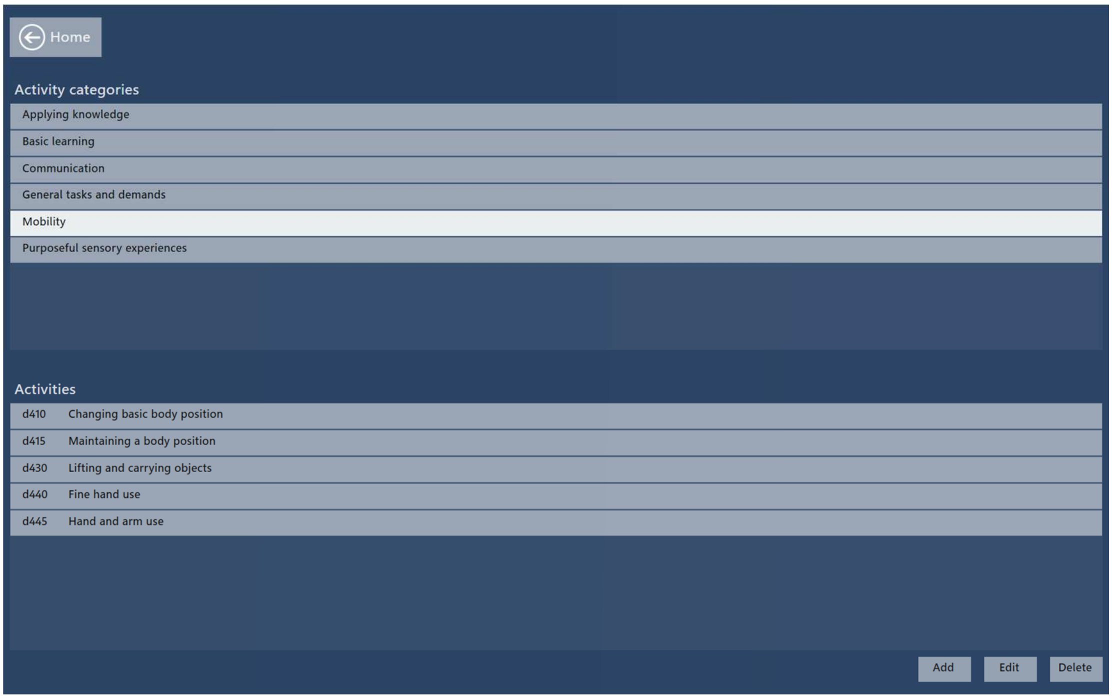


Figure 91: Bean Administration Suite (activities).

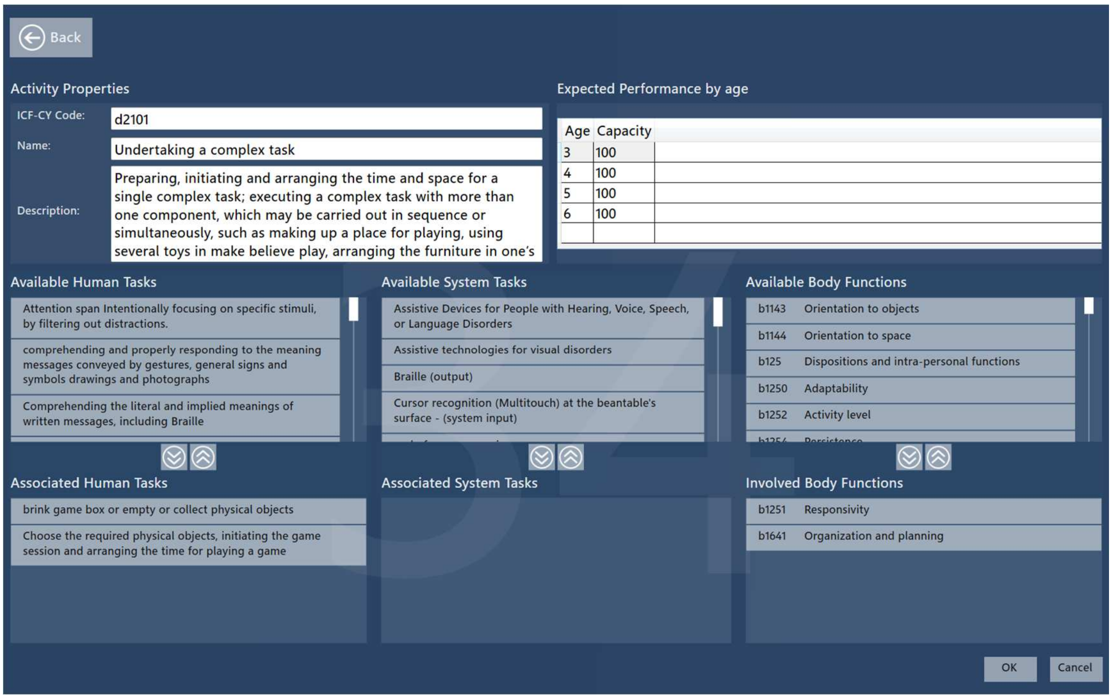


Figure 92: Bean Administration Suite (activity properties).

Early intervention professionals are able to add, edit or remove details for a smart game along with the required activities as depicted in Figure 93. The type (i.e., specific or general) of each required activity can be found in the right half of the same figure as along with the age range in which the activity is evaluated by the smart game.

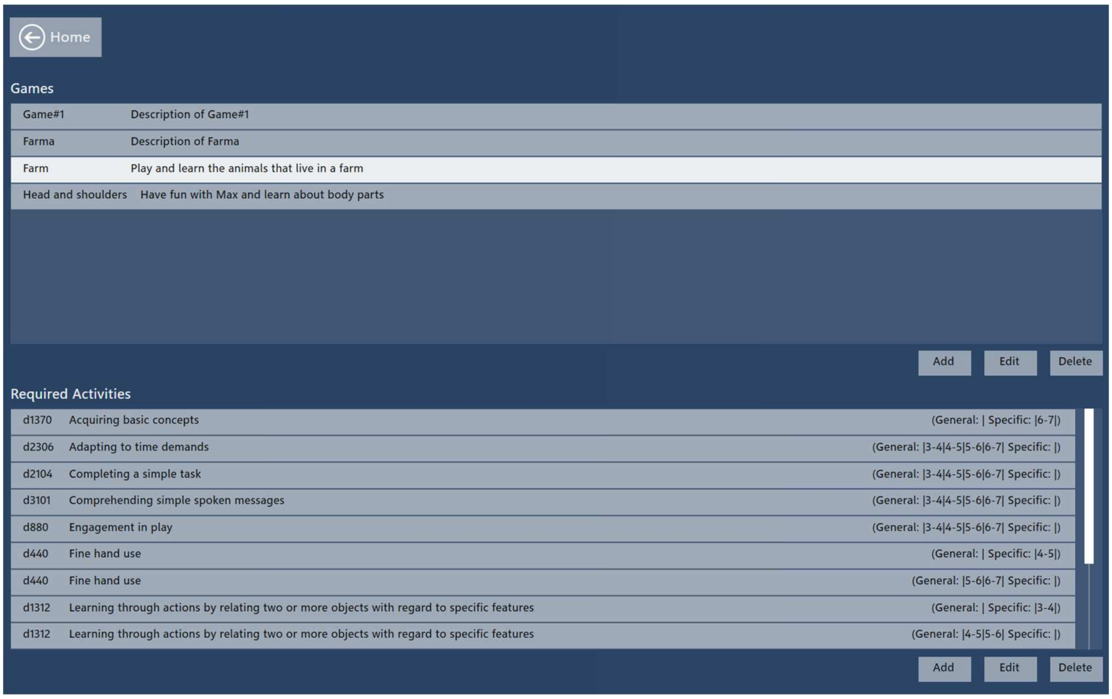


Figure 93: Bean Administration Suite (smart games and required activities).

Regarding a required activity, the author is able to select its type (specific or general) as well as the age range in which the activity is evaluated by the smart game (see Figure 94). The author has to select which the main activity is using the drop down list where all existing ICF-CY activities appear. In this way, the authors are able to enable only those human and system tasks that are relevant to the selected smart game. Optionally, they are able to rename them in order to give a description closer to the task expected to occur during the selected smart game.

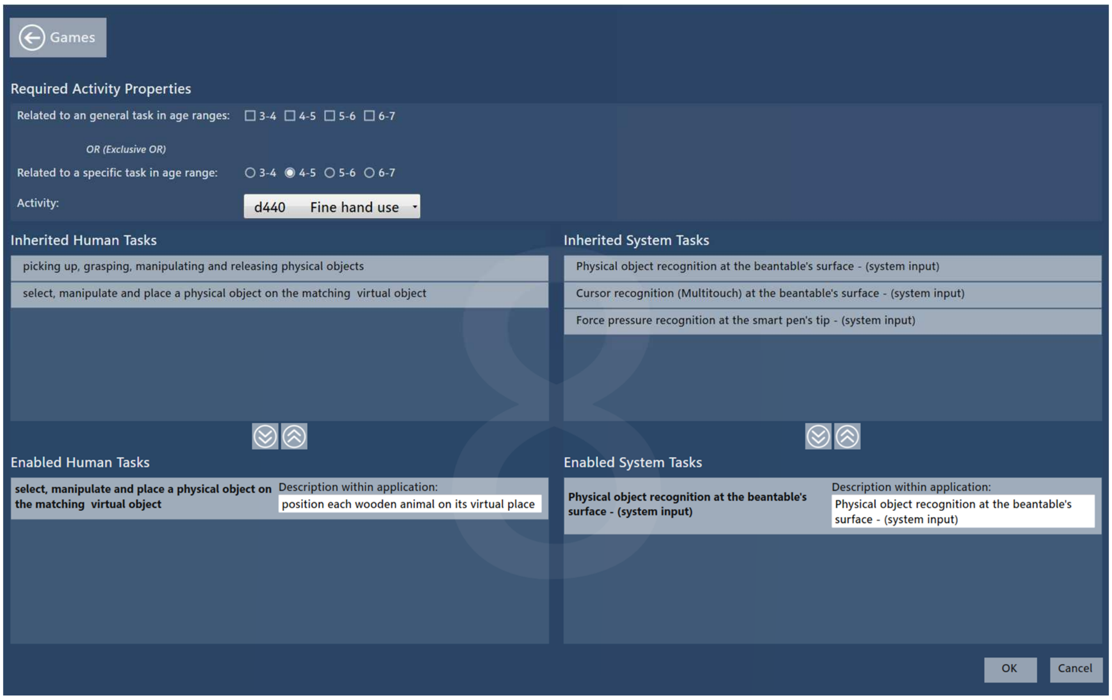


Figure 94: Bean Administration Suite (required activity)

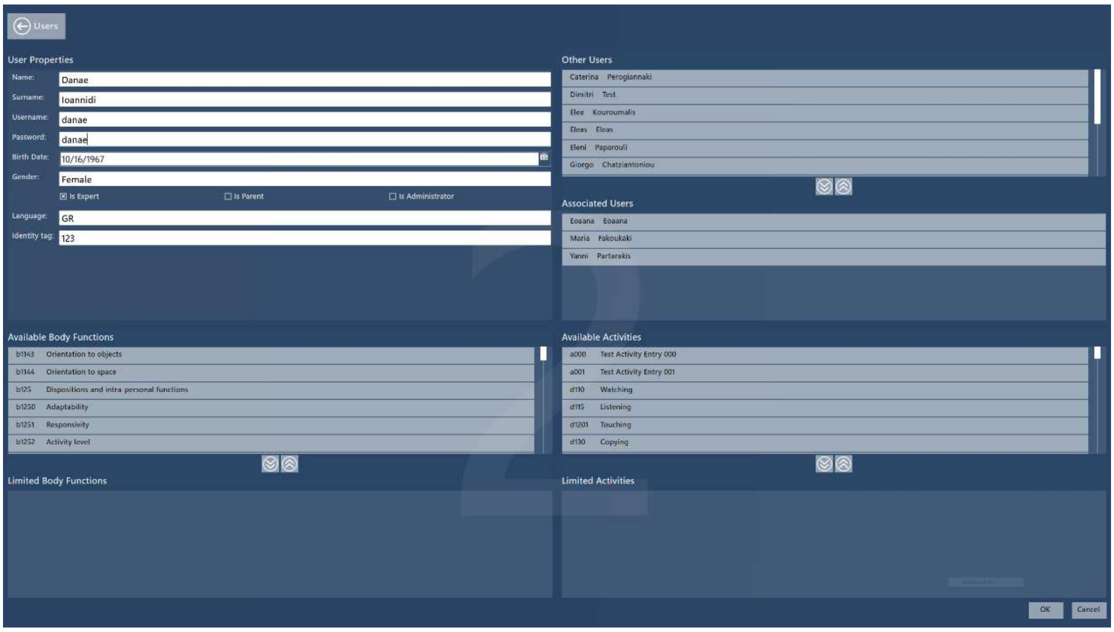


Figure 95: Bean Administration Suite (user properties)

System administrators are able to add, edit or remove users and set their role within the proposed framework (e.g., child, parent, child development expert or administrator). As depicted in Figure 95, early intervention professionals are able to configure a child's profile by choosing the activities in which the child has potential limitation. Additionally, experts can select those body functions in which the child has problems according to his clinical profile. System administrators are able to associate users for a selected user underlying that there is a relationship among them. This means that in case the selected user is parent, the associated users are his/her children, and similarly, in the case were the selected user is a child development expert, the associated users are his/her clients.

Figure 96: Bean Administration Suite (parents report)

4.10.4.2 Reporting child's play maturity (Parents)

Parents can access reports about their children's playing maturity development progress. As Figure 96 illustrates, parents can be informed if the measured maturity of their child responds to the expected one for their age and for the specific smart game. In detail, they can see in a bar chart (in the lower left of the figure) the child's actual age (light blue) compared with the age inferred (yellow) by the adaptation infrastructure mechanism (ADAM, see section 4.3). In the middle of the same figure, parents can see, through the pie chart (depicted in a percentage scale), how many times the child quitted the game or how many times the child completed the game successfully.

Parents are able to monitor in real time the child's maturity regarding the targeted required activities by selecting the "Refresh" button. To the left of the Figure 96, there are statistics about the a) total session number (i.e., times smart game started), b) total duration, c) number of successfully completed sessions, d) number of quitted sessions, e) gender, f) first and last time the child played the selected smart game, and g) child's name. Half way down of the same figure, there is a comparison bar chart for the last six months between the expected capacities of each required activity (light blue bars) and the capacities measured during interaction monitoring from the selected smart game in conjunction with ADAM. The same comparison bar charts is shown at the bottom right of the same figure with the difference that it has been calculated on the basis of all play performance scores during the entire interaction with the smart game.

Finally, as seen in the upper right of the figure, parents are provided with recommendations about what to observe during child's smart game playing. In detail, a recommendation is generated when the capacity of a required activity is much lower than the expected and follows a downward path. That recommendation consists of the name and the description of the required activity, as well as the corresponding human tasks. Parents are able to print or send via email the entire report including possible recommendations.

Figure 97: Bean Administration Suite (early intervention professionals' reports, screen 1/2, no smoothing)

4.10.4.3 Reporting child's play maturity (early intervention professionals)

Early intervention professionals can have access to further information about their client's playing maturity. Besides having access to the aforementioned reports designed for parents, they are also able to examine individual sessions for reasoning about whether the child meets the developmental milestones. At the left top of the Figure 97 there is a list of all recorded sessions for the selected smart game. Early intervention professionals are able to make multiple selections of only those sessions that are of interest to them by clicking in each of them separately or by using the available tools next to the list (e.g., selection within a period of time). Every time the selected sessions change, the remaining parts of the figure are updated. In the upper middle of the figure, there is a pie chart presenting the percentage of sessions in which smart game was successfully completed, as well as the percentage of sessions where the game stopped or closed automatically. At the top right, the recommendations area appears as before, with the only difference that now the required activity with low capacity is further characterized by its ICF-CY code and its involved body functions.

Figure 98: Bean Administration Suite (early intervention professionals' reports, screen 1/2, smoothing)

Early intervention professionals are able to monitor remotely and in real time the child's play performance by pressing the "Refresh" button or more simply having enabled the "Auto Refresh" option. Additionally, they might enable the smoothing option in order to see the line charts in a more comprehensive way (see Figure 98).

Every time new data arrive, the line charts are automatically updated. The first line chart presents interaction history of the child's play performance during the selected sessions in relation with the age related automatic adaptations that ADAM suggested to the selected smart game. The second line chart presents an analysis of the child's play performance using time series forecasting methods (i.e., weighted moving average) during the selected sessions (ADAM, see section 4.3 for further details about the employed analysis methods). The third line chart presents the time independent interaction history of the child's play during the selected sessions in relation to the age related automatic adaptations made by ADAM. Similarly, the performance depicted in this chart has been calculated using the aforementioned time series forecasting methods.

Early intervention professionals are provided with additional charts by choosing the "Next" button in the right middle of the Figure 98. As depicted in Figure 99, the first chart is a line chart presenting the child's capacity (based on smart game's committed play performance) in every required activity (provided that there are performance records within selected sections). As before, the same chart can be created using a more comprehensive way by enabling the smoothing option as seen in Figure 100. The second chart is a bar chart which presents information about the specific required activities of the selected smart game. In detail, it presents: a) the expected capacity (light blue color) of the required activity according to Denver II scale, b) child's capacity as the mean average of corresponding play performance scores (yellow color), and c) child's weighted capacity (pink color) which has been calculated using the time series forecasting methods (e.g., weighting moving average). Similarly, the third chart is a bar chart which presents information about, not specific but the general required activities of the selected smart game.

Figure 99: Bean Administration Suite (early intervention professionals' reports, screen 2/2, no smoothing)

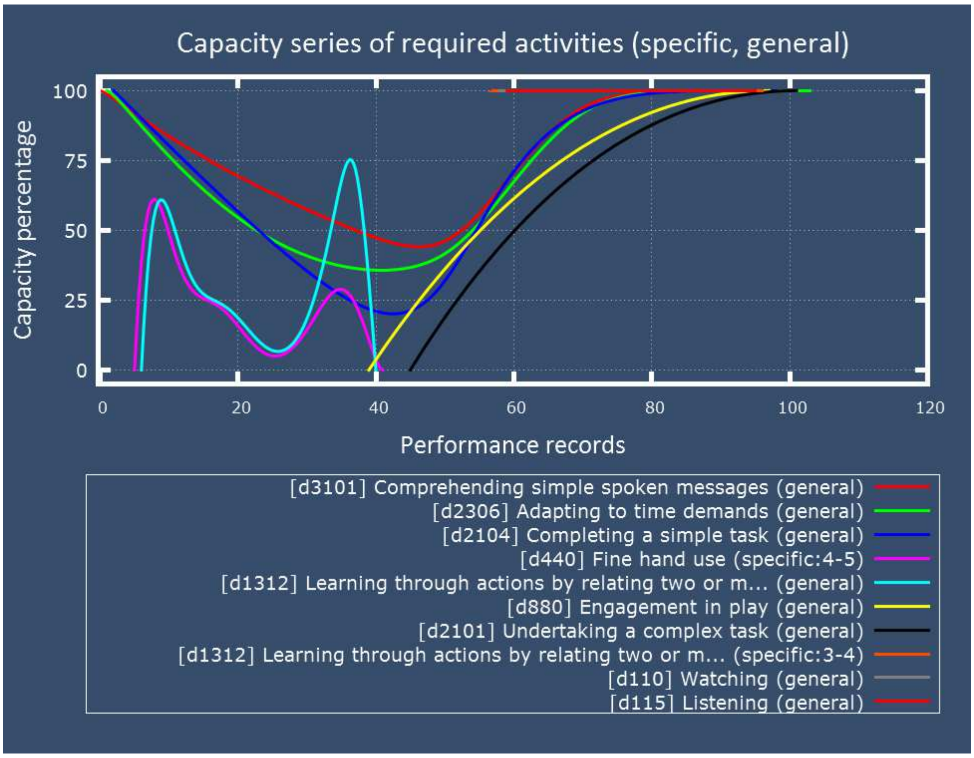


Figure 100: Bean Administration Suite (play performance series for each required activity, smoothing)

4.11 Deployment: the AmI playroom

The framework presented by this thesis has been deployed and tested in an AmI simulation space hosted within the AmI classroom simulation space of the FORTH-ICS AmI research facility (see Figure 101). In the ICS-FORTH AmI facility [212], various AmI technologies and applications are installed, integrated and demonstrated, and alternative ideas and solutions are cooperatively developed, studied and tested.

Figure 101: ICS-FORTH AmI facility

As depicted in Figure 102, the simulation space consists of: a) the augmented artifact Beantable accompanied by the augmented chair, b) the secondary display device presenting Max, and c) a large display TV (embedded PC running windows 8.1) presenting Max as well.

Figure 102: The extended Beantable setup

5 Case studies

The proposed framework provides the means to develop games that employ the environment and technology to offer novel play experiences to children. This section presents **feature demonstration games,** as well as traditional games turned into **smart games,** that can: a) monitor children's continuously changing abilities and consequently adapt to them, and b) act as assessment tools of children play maturity, providing early intervention professionals with relevant information about each activity occurred during playing (e.g., play performance).

5.1 Smart Games

Three popular games have been redesigned, implemented and deployed within an Ambient Intelligence (AmI) simulation space. Within this space, the presented augmented artifacts (e.g., Beantable), the sensory infrastructure (i.e., Nibbler), as well as the virtual character (i.e., Max) are employed to extend the purpose and objectives of the games, thus also expanding their applicability to the age group targeted by the proposed framework (i.e. , preschool children from 3 to 6 years old).

OT expertise has been employed for the design of the smart games, so as to meet the needs of OT common practice, but also for providing the knowledge at the basis of the monitoring and adaptation logic employed by the games. The games are capable of monitoring and following the progress of each young player, adapt accordingly and provide important information regarding the abilities and skills of the child and their development over time.

The developed smart games build upon child development theories and the definition of expected skills and tools (as discussed in sections 2.3 and 2.4). Activity analysis and play performance analysis were employed to design and shape the adaptation needs of the games, allowing them to evolve together with children' developmental requirements. Activity recognition is employed to monitor interaction during game playing.

5.1.1 The farm game

The farm game is tailored to the needs of preschool children in the age range targeted by the proposed framework, and supports playing through tangible interaction with physical objects. The design of the game was based on an actual physical puzzle game. The puzzle pieces are used to act as the physical part of the game, while the background is provided digitally on the Beantable's surface. Wooden pieces, including physical objects and the identity card, were scanned in order to create their virtual counterparts employed in the game. Special visual markers (i.e., fiducial symbols) were added on the bottom side of the physical pieces (see Figure 103 Left) which are recognizable by the system.

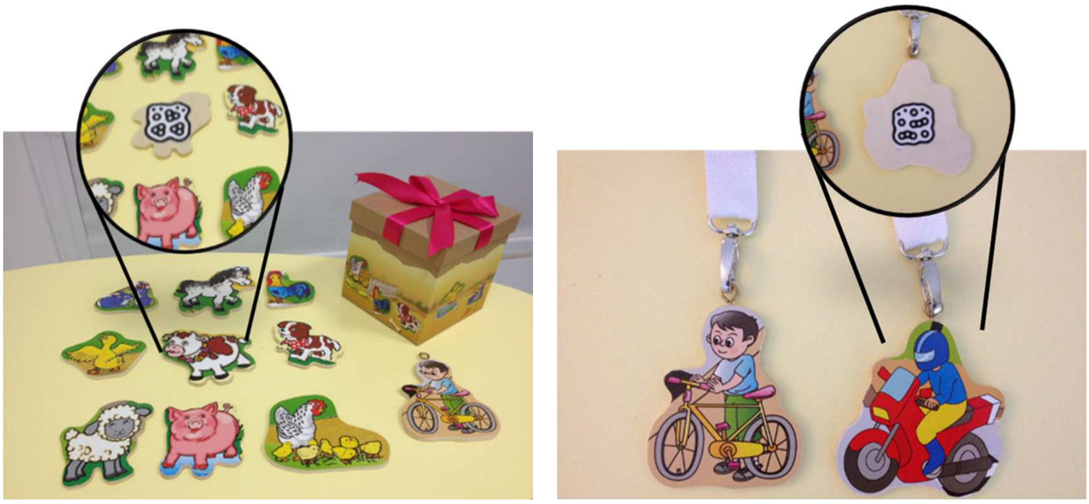


Figure 103: Physical items and attached visual markers (Left). Tagged "identity cards" with lanyards (Right).

Additionally, in the case of identity cards, lanyards were placed on the top of the physical objects as shown in Figure 103 (Right). Children store all the physical items in a box called the "Farm Box", which also has a visual marker attached.

Figure 104: Waiting for the identity card

Puzzles in general are an important learning tool for toddlers and young children, as they provide many skills and mental learning benefits and opportunities. Psychologists have determined that a child's brain development is influenced significantly when a child acts on or manipulates the world around him or her. Puzzles provide that opportunity as children learn to work directly with their environment and change its shape and appearance. Playing the Farm game, a child has the opportunity to demonstrate and practice skills such as:

- **Cognitive Skills**: The theme of the game is animals and their environment in different seasons and the puzzle may serve as the medium for learning the animals.
- **Fine Motor**: In a funny way children can develop and refine their fine motor skills since when engaged in playing with the farm game they need to pick up and grasp pieces and move them around, manipulating them and fitting them into the correct places.
- **Hand and Eye Coordination**: Playing the game requires a trial and error process which involves hand and eye coordination.
- **Setting Small Goals**: As a child works on the puzzle, he or she has to develop a strategy to work the puzzle faster and more efficiently, thus learning to achieve small goals as a means toward a larger goal.
- **Problem Solving**: Completing a level sets a single goal to achieve. Children must think and develop strategies towards achieving this goal. This process involves problem solving, reasoning skills and developing solutions which can later be transferred to personal/adult life.
- **Self Esteem**: Completing a level brings considerable satisfaction to a child. Overcoming the challenges involved in solving a puzzle gives them a sense of achievement and pride that provides a lift to their self-confidence and self-esteem.

5.1.1.1 Farm game activity analysis

The game requirements were elaborated taking into account that the ways and methods employed for child's interaction with the system can influence child's performance. The requirements were collected and categorized as follows.

Context demands are the components that form the environment where the game is played and may influence the player's performance. The p**hysical environment** consist of a wooden desk and chair (**Beantable**), where the child should remain seated while matching wooden pieces with corresponding places on a virtual game board. The v**irtual environment** contains: a) still and animated images, b) tangible objects, c) virtual representations of the physical objects, and d) avatar's speech dialogues and biped animations.

Play demands are the equipment and materials needed for the game which are: a) an identity card, and b) the Farm Box which contains 9 wooden animal puzzle pieces.

Player factors include the social and performance skills needed for play according to the age range that the game is proposed for. The player has to execute a number of tasks for achieving the game goals or demonstrate behaviors related to activities such as watch, listen, adapt to time demands and more, and also to communicate with the virtual character **Max**.

Body Functions According to the ICF–CY, the related physiological and psychological functions for the execution of playing actions were identified. The selected body functions used to support the design rationale include among others:

- Mental functions involved in discriminating shape, size, color and other ocular stimuli
- Mental functions involved in distinguishing by sight the relative position of objects in the environment
- Mental functions involved in discriminating sounds and other acoustic stimuli
- Specific mental functions for recognizing and using signs, symbols and other components of a language
- Statesthesia and Kinaesthesia
- Mental functions involved in acquisition of knowledge about objects, events and experiences; and the organization and application of that knowledge in tasks
- Mental functions that produce complex goal directed sequences of movement
- Attention, thought, problem solving, time management, organization and planning functions
- Activity level, voice and speech functions.

Performance skills are a set of activities and the underlined skills that should be used by a child in order to play the game efficiently. This set was also used for recording the capacity and play performance of children's abilities during interaction:

- Remain seated in front of the desk
- Listen to Max spoken messages and respond properly
- Respond to system instructions on time
- Place and empty the game box on the Beantable's surface or collect physical objects
- Locate and observe interactive animal virtual pictures
- Select, manipulate and place a physical object on the matching virtual object
- Listen to animal sounds
- Comprehend and respond to the messages conveyed by visual stimulus, drawings and pictures
- Initiate the game session, choose the required physical objects, and manage the time for playing the game
- Complete the session by placing all wooden animals on their corresponding virtual pictures without assistance.

5.1.1.2 Starting point of the game

The game is started through the Beantable's startup screen by selecting the image of the "Farm". At the beginning of this game, Max asks the child to place his identity card on the surface as shown in Figure 104. The system recognizes child's identity card and remotely requests his profile from the adaptation infrastructure mechanism (ADAM). Using the profile, the system initializes the game accordingly, and Max welcomes the child with his/her name.

Figure 105: Waiting for the "Farm Box"

At this point, Max asks the child to find and place the "Farm Box" on the Beantable's surface as shown in Figure 105. Thereafter, the game will be started at the level corresponding to the child's profile, while Max explains the relevant instructions. If no action is performed, after waiting a period of time, Max says a "Good bye" message, and the game is terminated.

5.1.1.3 First Level (from age 3 to 4)

At the first game level, the child has to place the physical objects from the "Farm Box" in the correct virtual positions on the digital game board presented by Beantable (see Figure 106). Each virtual position is presented in the form of a black and white animal's picture. The child can remove or leave the physical objects on the surface without getting any feedback (see Figure 108 A). The employed **PhysicalObjectRecognition** module (as presented in 4.9.3) recognizes the matching between the physical object and its virtual position as a success, even if the object is not properly oriented. When a match is achieved, the corresponding image of the animal on the virtual game board is colored (see Figure 108 B). For each successful matching the system rewards the child with the reproduction of the animal's sound.

While playing, the system counts the errors occurred for each virtual position. These errors, as well as child's progress, are continuously reported to the adaptation infrastructure mechanism (ADAM) for analysis as described in section 4.3. In this context, error means placing any incorrect item in a specific virtual position. When the errors of a specific virtual position are more than 3, the corresponding animal sound is produced. If the child tries to place a wrong physical item in a virtual position that was previously successfully completed, the related sound is activated, and this is also calculated as an additional error. On the contrary, every time that the child places a physical item in the correct position, which has been completed successfully in the past, the sound for this item is activated as well. When all the physical objects are placed successfully the game round is finished. When finishing the game at any of the available levels, two phases are initiated**:**

- **Congratulation phase**: At this point, Max congratulates the child by saying a random message such as "Congratulations", "Bravo", "Very well", etc. The system also informs the adaptation logic about the end of this round.
- **Free play mode**: At the end of the "congratulation phase", the free play mode starts. During this mode, the child can interact with the system without errors being signalled. In that case, the system considers successful any movement that results to a correct matching of physical objects to every virtual position on the game board. In this way, the child can create his own play situation and enjoy the game undisturbed. When the child stops interacting with the game for a few seconds, this mode is disabled.

Figure 106: An indicative instance of the first level

After the completion of the aforementioned stages, the system checks the presence of physical items on the surface, and Max requests the child to collect the remaining physical items from the table uttering the message "Please, put the animals in the box". If the child reacts promptly, the system proceeds to the next step. Otherwise, the system will close the game. After the completion of the previous task, Max produces the message "Will you continue?" At this point, the system is waiting for speech input. If the child answers "Yes", the system will start a new round at a level corresponding to the child's abilities which are inferred by the game's adaptation logic.

Figure 107: An indicative instance of the second level

Figure 108: Alternative instantiations of virtual animals

5.1.1.4 Second Level (from age 4 to 5)

The system has similar functionality as in the first level. At the second level, the child has to place more physical items in their virtual position (see Figure 107). Moreover, the game logic is extended to account for proper object orientation recognition. In case of correct matching, the black and white image of the animal is turned into colored following by its sound.

During the game, if the child faces significant difficulties to properly match the game items, the adaptation logic (which monitors continually the game interaction) informs the system that the child has to play at a lower level for the rest of the round. Thereafter, the child will continue the game playing in free play mode (without errors being signalled). In case that the child prefers to play again, the system calls the adaptation logic to get information about the starting level of this round.

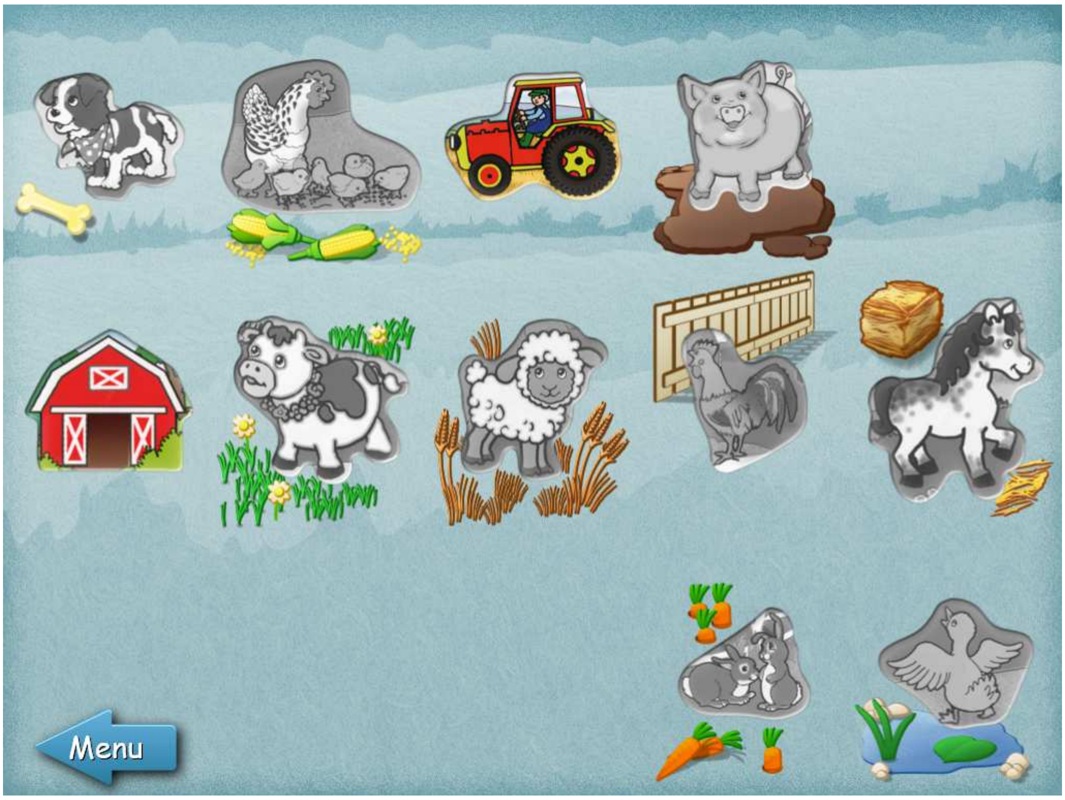


Figure 109: An indicative instance of the third level

Figure 110: An indicative instance of the fourth level

5.1.1.5 Third Level (from age 5 to 6)

At this level, a round of the game is considered completed when the child has placed all the physical objects on the virtual positions and properly oriented (see Figure 109). It is important to mention that the physical items must remain in their virtual positions until the end of this round. When the child places all the physical objects in the virtual positions, the system will check the matching of these items. If the child has placed successfully all the physical objects at their corresponding locations on the interface, the round of this game is completed and the system starts the “Congratulation phase” followed by the “Free play mode”.

5.1.1.6 Fourth Level (from age 6 to ~7)

At this level, the functionality remains exactly the same as at third level, but the board of the virtual positions is created using the mirrored and the normal outline of each physical object (see Figure 108 C, D). An example of a game round at the fourth level is shown in Figure 110. A round of the game is considered completed when the child places the required physical objects at their corresponding virtual positions properly.

5.1.2 Tower game

The tower game is tailored to the needs of preschool children in the age range targeted by the proposed framework, and supports playing through tangible interaction with augmented artifacts (see Figure 113). The tower game allows children to learn, identify and compare six different colors on each side of the dice with those illustrated on the path of the game (see Figure 111). Similarly, at the advanced level, the user is called to recognize the numbers on the dice and depict its value while moving over the path from tile to tile. During playing, children meets a variety of animal figures, trees and other virtual objects. Used augmented artifacts, including digital dice and augmented pen, are stored in a box called the “Tower Box”, which has a visual marker attached.

Regarding the gameplay, the child rolls the colored dice and moves forward until he reaches the first tile of the same color. In the case of a numbered dice, the child must recognize the number that he rolled and move forward that number of spaces. Two sets of numbers are used, 1-5 and 6 -10 according to level status. If his action is correct, he rolls the dice again and continues the previous procedure. If the dice becomes white, a card appears; this card has a random position on the path (either back or forth of the current position) which the user should move to. If the user reacts correctly, the card will be added to a staple of similar cards next to the display. The last tile of the path represents the entrance into the tower, however if the player rolls too high, he then has to move backwards. The child can only win by rolling the exact color or number needed to land in the tower.

The game round is finished when the child lands at the tower and has already selected cards. On the other hand, if the child lands at the tower but some cards are missing, Max asks him to roll again and explains him the reason why (i.e., “You haven’t selected all of your cards, throw the dice again”).

Tower Game is a simple maze board game. It has simple rules and it offers opportunities to promote knowledge, and practice fundamental skills such as color and number recognition, counting and writing. During play the child can acquire knowledge of concepts such as seriation, counting and ordering and at the same time practicing fine hand use and other skills related to numeracy and the use of writing implements. He also learns to integrate sets of actions so as to follow rules and to sequence and coordinate his movements.

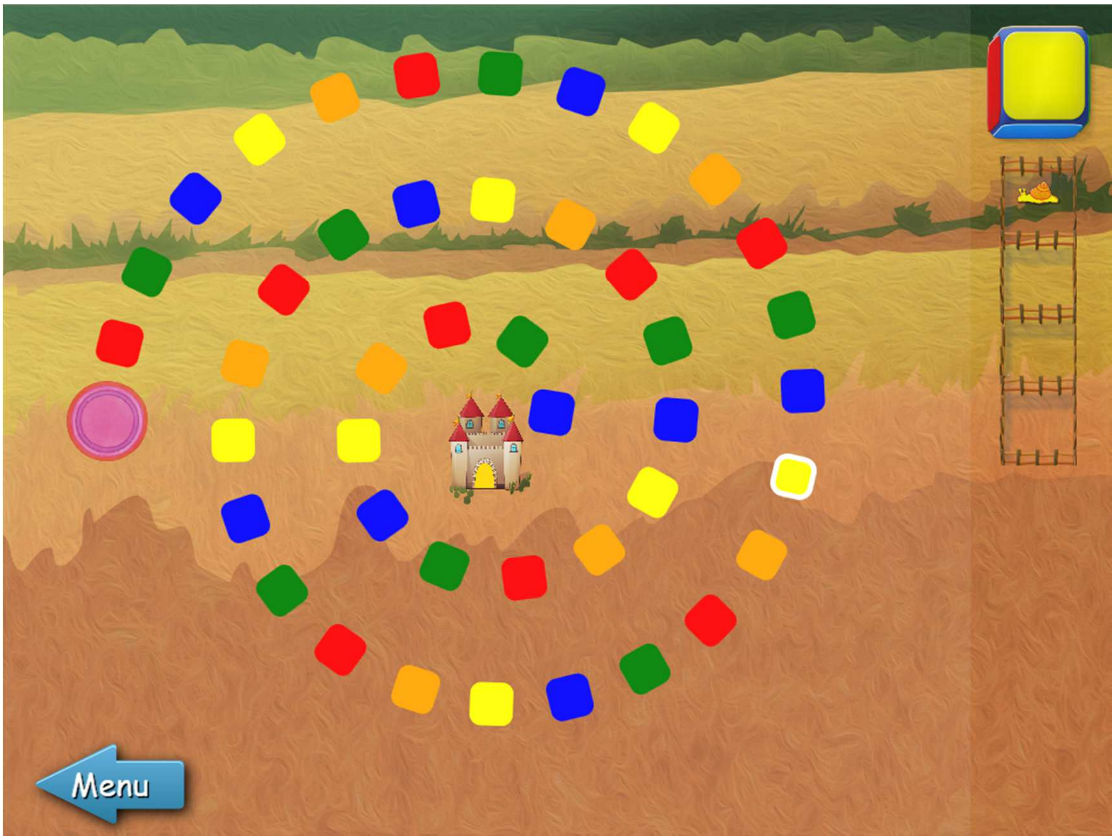


Figure 111: Tower game

5.1.2.1 Tower game activity analysis

Game requirements were elaborated taking into account that the ways and methods employed for child's interaction with the system can influence child's performance. The requirements were collected and categorized as follows.

Context demands are the components that form the environment where the game is played and may influence player's performance. The p**hysical environment** consists of a wooden desk (**Beantable**) and a chair (augmented chair), where the child should remain seated while playing. The smart die and the augmented pen are also assumed as components of the physical environment. The v**irtual environment** contains: a) still and animated images, b) tangible objects, c) virtual representations of the playing cards, and d) virtual avatar's speech dialogues and biped animations.

The equipment and materials demanded for playing the game are: a) an identity card, and b) the "Tower Box" which contains a die and the augmented pen.

The social and performance skills needed for playing each level of the game are defined according to the child's age. For achieving the game goals, the player has to execute a number of tasks or demonstrate behaviors such as watch, listen, adapt to time demands and more, and also to communicate with the virtual character **Max**.

Body Functions that have been identified are related to physiological and psychological functions. The selected body functions used to support the design rationale include among others:

- Mental functions involved in discriminating colour and representation of symbols, numbers, pictures and other stimuli perceived through vision.
- Mental functions involved in visual spatial perception
- Mental functions involved in acoustic stimuli
- Mental functions involved in gaining knowledge about the use of objects,; and the organization and application of that knowledge in tasks
- Mental functions that produce coordinated and targeted movements
- Mental functions related to numeracy and writing
- Higher level mental functions related to attention, thought processing, decision making, time management, space orientation, organization and planning functions
- Psychological functions related to activity level
- Voice and speech functions.

For recording the play performance and player's abilities during interaction, selected activities have been used for elaborating the **skills** needed for to play the game efficiently. For the Tower game, the performance skills include:

- Remain seated for a required period of time
- Listen to spoken messages and respond properly
- Respond to system instructions on time
- Use the objects in the game box on
- Locate and observe interactive virtual tiles of the maze and other pictures or visual elements
- Performing the coordinated actions of handling the smart die or the augmented pen
- Point an index finger and onto virtual tiles in proper direction
- Use a writing tool (augmented pen)
- Learning to use such concepts as colours and numbers
- Learn to integrated sets of actions so as to follow rules and to sequence and coordinate movements, intentionally maintaining attention to specific actions or tasks for an appropriate length of time
- Comprehend and respond to the messages conveyed by visual stimulus, drawings and pictures
- Ordering and counting
- Undertaking and completing the required tasks
- Handling stress.

The conditions for successful user performance include:

- The tiles touched by the child must be literally the one after another either forward or backwards.
- The first tile after every throw must be the next to the tile that the user landed last.
- The system considers automatically that the child has completed a turn after a few seconds waiting on one tile.
- According to the above, the system compares the path followed by the child with the best estimated path as knowing the start and the end position.

If the child chooses to touch wrong tiles, these are highlighted in red, so that he can recognize when he makes a mistake.

5.1.2.2 Starting point of the game

The game is started through the Beantable's startup screen by selecting the image of the "Tower". At the beginning of this game, Max asks the child to place his identity card on the surface as shown in Figure 112 (Left). The system recognizes the child's identity card and remotely requests his profile from the adaptation infrastructure mechanism (ADAM). Using the profile, the system initializes the game accordingly, and Max welcomes the child with his/her name.

Figure 112: Max asks the child to place his identity card (Left). Max asks the child to find and place the "Tower Box" on the Beantable's surface (Right).

At this point, Max asks the child to find and place the "Tower Box" on the Beantable's surface as shown in Figure 112 (Right). Thereafter, the game will be started at a level corresponding to the child's profile, while Max explains the relevant instructions. If no action is performed, after waiting a period of time, Max says a "Good bye" message, and the game is terminated.

5.1.2.3 First Level (from age 3 to 4)

At the first game level, the child has to move over the path-maze by touching the tiles one after another with his finger until he reaches the first tile with the color that the dice shows (see Figure 113 - Left). The maximum number of different positions is six and the dice used at this level illustrates only colors. In detail, the user throws the colored dice and touches with his finger one tile at a time until he reaches the right one. As the game begins, in case the child has not reacted to the system's instructions in order to start playing (i.e., "Choose a piece and throw your dice"), the first tile of the path starts blinking followed by an appropriate voice message. If a dice-throw goes white, the player has the opportunity to save an animal that appears each time at a different position by moving forwards or backwards to that position.

When finishing the game at any of the available levels, two phases are initiated**:**

- **Congratulation phase**: At this point, Max congratulates the child by saying a random message such as, "Congratulations", "Bravo", "Very well", etc. The system also informs the adaptation logic about the end of this round.
- **Free play mode**: At the end of the "congratulation phase", the free play mode starts. During this mode, the child can interact with the system without errors being signalled. The child can freely roll the dice and interact with the game. In this way, the child can create his own play situation and enjoy the game undisturbed. When the child stops interacting with the game for a few seconds, this mode is disabled.

5.1.2.4 Second Level (from age 4 to 5)

The system has similar functionality as at the first level. The child has to move over the path by *dragging* his finger over the tiles one after one as writing a line until he reaches the color-name. During the game, if the child faces a significant difficulty or makes wrong movements, the system repeats the color name in order to help him to move on. After three wrong attempts the game terminates. Following the same concept as at the first level, when the child drags his/her finger over the tiles they lighten for a few seconds. While entering the tower, a different animal appears each time.

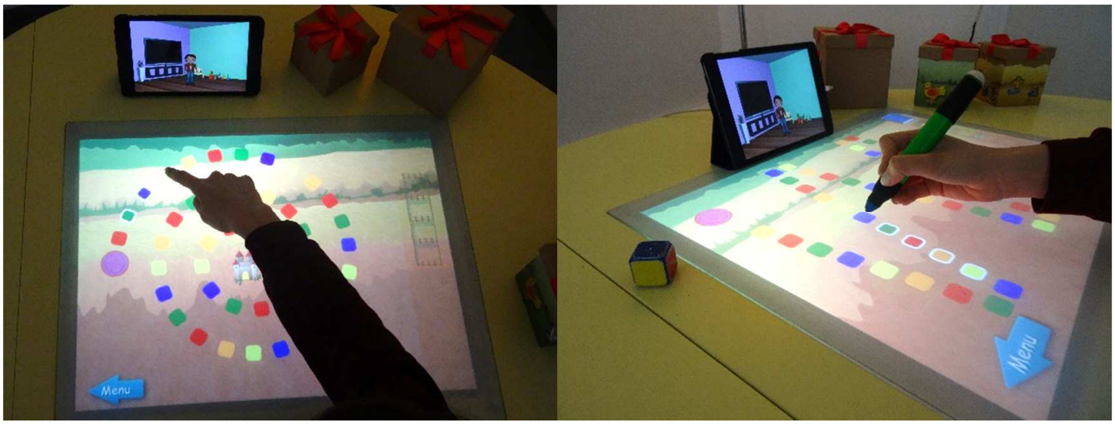


Figure 113: Various interaction techniques required during playing with Tower game

5.1.2.5 Third Level (from age 5 to 6)

At this level, the child has to move over the path as before by *dragging* his/her finger over the tiles. The difference between the previous level and the current one is that at this level the child uses a dice with numbers and not with colors. After rolling the dice, the system reads the number and instructs the child to count as many tiles on the path in order to land on the tile that corresponds to that number. While the child drags his finger over the tiles, they lighten as long his movements are correct.

5.1.2.6 Fourth Level (age 6 to ~7)

At this level, the functionality of the system remains exactly the same as at the previous levels, apart from the fact that the child has to move over the path by dragging a *augmented pen* over the tiles as writing a line in a maze (see Figure 113 - Right). A round of the game is considered completed when the child lands into the tower icon. Afterwards, the system starts the "Congratulation phase" followed by the "Free play mode".

5.1.3 Mimesis game

The Mimesis game is an entertaining game for preschool children in the age range targeted by the proposed framework. The game requests the child to assume various body postures as illustrated in Figure 114. Using the Nibbler sensory infrastructure, the game measures the quality and performance of the body posture that the child assumes and can extract indications of the achieved maturity level and skills of the child. The Mimesis game was scripted in ACTA by early intervention professionals (see APPENDIX III – Mimesis game: ACTA script). In detail, the child stands in front of the interactive display, which is positioned horizontally, (e.g., near to a wall) giving the feeling of standing in front of a mirror. At the upper side of the large display, a depth camera (i.e., Kinect sensor) is positioned, which allows Nibbler to recognize the user position and his gestures as well. The sensor sets restrictions regarding the view area that ranges from 40 cm to 4 m and in around 45 degrees viewing angle. As a result, Max uses verbal commands to ensure that the player remains within the camera restriction area for successful interaction with the game. For example, in case the user comes very close to the display or exits the view area of the sensor, Max advices him with a message to move towards the right position (e.g., "Move back so that I can see you better").

The Mimesis game requires the child to imitate a series of postures demonstrated by the system, a process that requires motor planning or programming a skilled, non-habitual motor act. Moreover when verbal commands are used, the child's abilities related to understand and follow verbal directions and to plan movements based on verbal instructions are also stimulated. Posturing is a specific mental function that requires control over both motor and psychological events at the body level which results in an intentionally or habitually assumed position. The Mimesis game promotes the child's posturing ability by challenging him/her to imitate a series of demonstrated postures. According to Jean Ayres theory [13], imitation of postures requires higher level of sensory integration, based on assimilation of sensation from several sensory modalities and requiring greater central nervous system processing. Since postures presentation by Max is via ocular and audio stimuli, other physiological and psychological Body Functions are also involved during playing this game namely: a) Mental functions involved in the reception of language for decoding messages in spoken, written or other forms, to obtain their meaning, b) Mental functions of Visual perception, and c) Proprioceptive functions for the perception of body position and movement (kinesthesia). The child has to imitate a number of demonstrated postures, activity that requires adequate copying skills.

The Mimesis game consists of the following functions: a) Max presenting various poses, b) Max assuming various postures (moving from his idle position to a posture), c) voice commands of each pose, and d) audio description of each pose.

Figure 114: Postures demonstrated by Max in the Mimesis game

5.1.3.1 Mimesis game activity analysis

The game requires the child to assume (imitate) a series of body positions demonstrated by the system. Such process requires the following body functions:

ICF-CY code	ICF-CY short description
b147	Psychomotor functions – posturing (motor planning or a skilled or

	non-habitual motor act[80])
b1670	Reception of language
b1561	Visual perception
b260	Proprioceptive function
b760	Control of voluntary movement functions

Table 15: Required body functions in Mimesis game

5.1.3.2 Starting point of the game

The game is started through the Beantable's startup screen by selecting the image of the "Mimesis". At the beginning of this game, Max asks the child to place his identity card on the surface as shown in Figure 104. The system recognizes the child's identity card and remotely requests his profile from the adaptation infrastructure mechanism (ADAM). Using the profile, the system initializes the game accordingly, and Max welcomes the child with his/her name.

At this point, Max asks the child to stand up and approach the large display next to the Beantable. Thereafter, the game will be started at a level corresponding to the child's profile, while Max explains the relevant instructions. If no action is performed, after waiting a period of time, Max says a "Good bye" message, and the game is terminated.

When the child enters the camera view area, Max welcomes the child by saying "Hello. I'm Max. Do you want to play?". If the child answers YES, Max asks from the child to imitate a pose by saying "Do what I do as fast as you can". Depending on the child's profile, Max uses one of the four given ways of assistance as described in the following levels. The first two are exclusively visual, while the second two are pure voice commands.

If the child answers NO, then Max looks sad and says "Goodbye". If the child does not react to the question, after a while Max welcomes him again. If the child moves away from the camera view area, Max says "You are leaving" and then looks sad and says "Goodbye". If the child returns after a period of time, Max welcomes again.

5.1.3.3 First Level

At the first game level, Max demonstrates a randomly selected static pose without giving instructions (verbal information). For example, Max puts his hand on his nose instantly without tracking his movement and without describing what he does. If the child reacts properly within a certain period of time, Max congratulates the child by saying a random message such as, "Congratulations", "Bravo", "Very well", etc. Thereafter, the game continues as presented in "Playing with Max" section (see 5.1.3.7). In the case that the child does not respond properly within a given period of time, the game proceeds to the second level. Finally, the game informs the adaptation logic mechanism (ADAM) about child's play performance.

[80] Information: page 5. Ayers A.J, Southern California Sensory Integration Tests Manual. Western Psychological Services. Los Angeles 1980

5.1.3.4 Second Level

At the second game level, Max repeats the gesture in which the child did not respond properly at the first level, without giving instructions (verbal information). For example, Max puts his hand on his nose while the child can track his movement. Similarly, Max does not describe what he does. If the child reacts properly within a given period of time, Max congratulates the child by saying a random message such as, "Congratulations", "Bravo", "Very well", etc. Thereafter, the game continues as presented in "Playing with Max" section (see 5.1.3.7). If the child does not respond properly within a given period of time, the game proceeds to the third level. Finally, the game informs the adaptation logic mechanism (ADAM) about child's play performance.

5.1.3.5 Third Level

At the third game level, Max demonstrates the gesture in which the child did not respond properly at the second level, with instructions (verbal information). For example, Max puts his hand on his nose instantly while describing what he does. If the child reacts properly within a given period of time, Max congratulates the child by saying a random message such as, "Congratulations", "Bravo", "Very well", etc. Thereafter, the game continues as presented in "Playing with Max" section (see 5.1.3.7). In the case that the child does not respond properly within a certain period of time, the game proceeds to the forth level. Finally, the game informs the adaptation logic mechanism (ADAM) about child's play performance.

5.1.3.6 Forth Level

At the fourth game level, Max imitates the gesture in which the child did not respond properly at the third level, with a verbal command prompting the child to follow. For example, Max puts his hand on his nose while the child can track his movement and he says "put your hand on your nose". If the child reacts properly within a given period of time, Max congratulates the child by saying a random message such as, "Congratulations", "Bravo", "Very well", etc., If the child does not react properly within a the set period of time, the game continues as presented in "Playing with Max" section (see 5.1.3.7) with the exception that if it is the 3rd consecutive time that the games passes to the fourth level the smart game ends with Max saying "goodbye". Finally, the game informs the adaptation logic mechanism (ADAM) about the child's play performance.

5.1.3.7 Playing with Max

The Mimesis game randomly selects a different posture and the procedure continues from the first level. Additionally, if the child leaves the area of the Nibbler, Max asks him if he leaves and in case of not returning back, the game ends with Max saying "goodbye". When all postures have appeared the game ends with Max saying "thanks" to the player and starting the presentation of a short compilation video of photographs taken during playing (see Figure 115 for some screenshots).

Figure 115: A short compilation video of photographs taken during Mimesis game

5.2 Feature demonstration games

This section presents a set of games aiming to simply demonstrate the features of the proposed framework regarding the co-existence and cooperation of the provided augmented artifacts. This set consists of a) a multi-touch memory game which employs a custom made stamp, b) jigsaw puzzles which involve physical objects, c) paint application using the augmented pen, and d) a classic labyrinth game in which children are able to control a virtual steel ball by tilting either the Beantable's surface or the augmented chair's sitting area.

5.2.1 Pick & Match

Memory games are an optimal way to stimulate a child's brain and help enhance mental power and strength. **Pick & Match** is a memory game with virtual cards where children are intended to locate the same pairs of cards within a collection of initially hidden cards. Cards are dynamically populated when the game starts. The game consists of different levels where each one has more cards than the previous. An example of the first game level is presented in Figure 116 (left).

Figure 116: The memory game Pick & Match (Left). Interaction with custom made stamp (Right).

Pick & Match allows the child to interact either by using fingers in the context of natural interaction (multi-touch) or by using a custom made stamp (see Figure 116 (right)), which is appropriate for very young children with difficulties using multi-touch devices. Auditory feedback is used in this game as well.

5.2.2 Playing jigsaw puzzles

Puzzle games are not only entertaining, but can also provide a variety of learning opportunities. According to [139], this type of learning is categorized as the "Eureka factor". To this end, two jigsaw puzzles composed of physical parts were developed and tested with young children. As depicted in Figure 116, the implemented games are: a) “Winnie the Pooh”, and b) “The Three Little Pigs”.

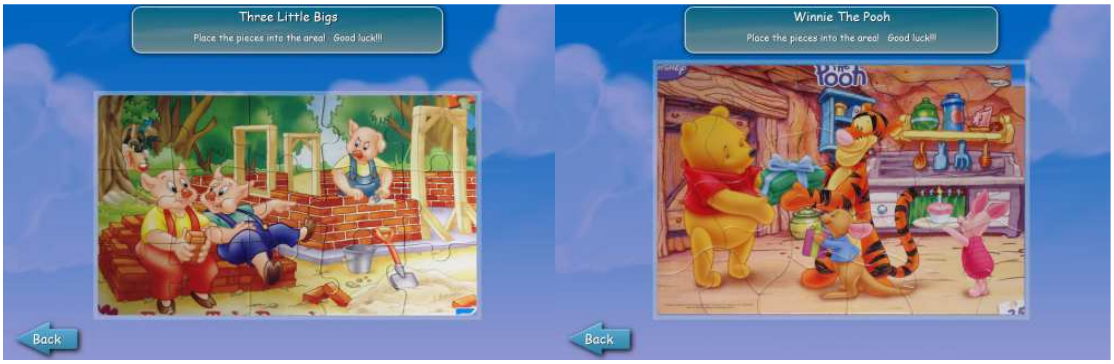


Figure 117: Jigsaw puzzle games: Three Little Pigs (Left) and Winnie the Pooh (Right).

While playing, the game is able to identify the location and rotation of any piece of the puzzle on the surface and provide scalable and personalized guidance through Max. For the moment, auditory feedback is provided upon each successful match between two pieces and the game ends with applauses when the child fully completes the puzzle. Furthermore, visual information is provided about the completion progress of the puzzle game.

Figure 118: Playing with jigsaw puzzle.

The puzzle game was developed in .NET framework using the C# programming language and WPF. Furthermore, it uses the **PhysicalObjectRecognition** module provided with the Bean SDK (as presented in 4.9.3). Each puzzle game was created using the authoring tool for tangible games presented in 4.10.1. The outcome of the tool is given as input to the **PhysicalObjectRecognition** module. Based on that, the puzzle game may help children to match a puzzle piece by highlighting the outline of its adjacent pieces.

5.2.3 Paint using the augmented pen

A paint application was developed as a usage example of the augmented pen’s sensing capabilities. Children can select ink colour using a colour wheel and freely draw lines using the augmented pen. As shown in Figure 119, children are able to automatically adjust the size of the e-ink by exerting more or less pressure to the Beantable’s writing surface. Additionally, children are able to erase part of the picture just using the augmented pen upside down.

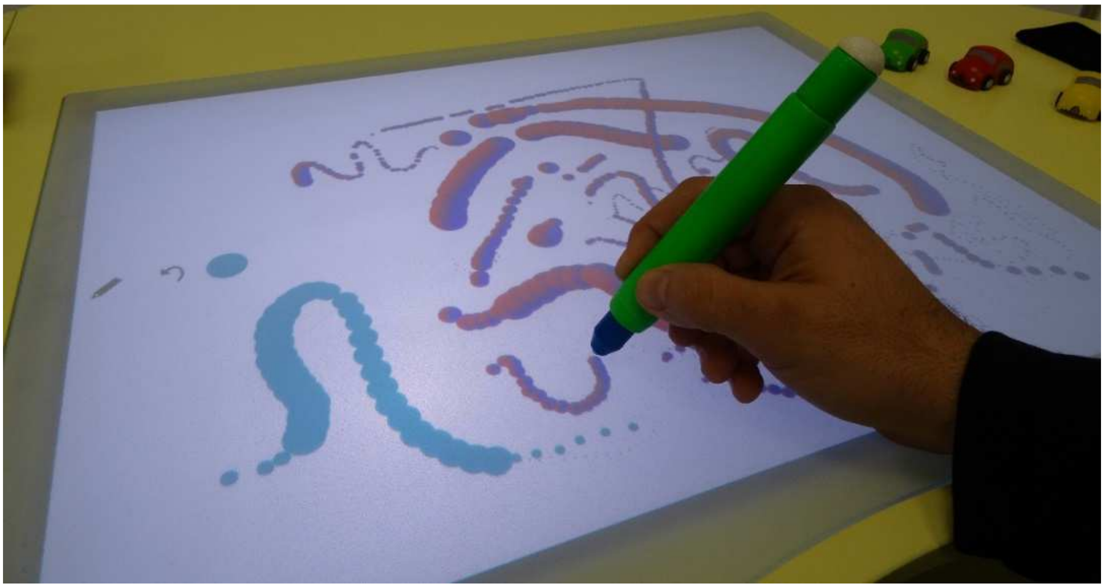


Figure 119: Paint application using the augmented pen

5.2.4 Labyrinth: a much different classic game

Labyrinth is a balanced bowling adventure game in which children are able to control a steel ball. Even though the gameplay is very simple, the game requires skills of physical mixing in order to get the ball to reach the destination. The game consists of various levels of difficulty as depicted in Figure 120. Each one has random obstacles blocking the forward movement of the ball.

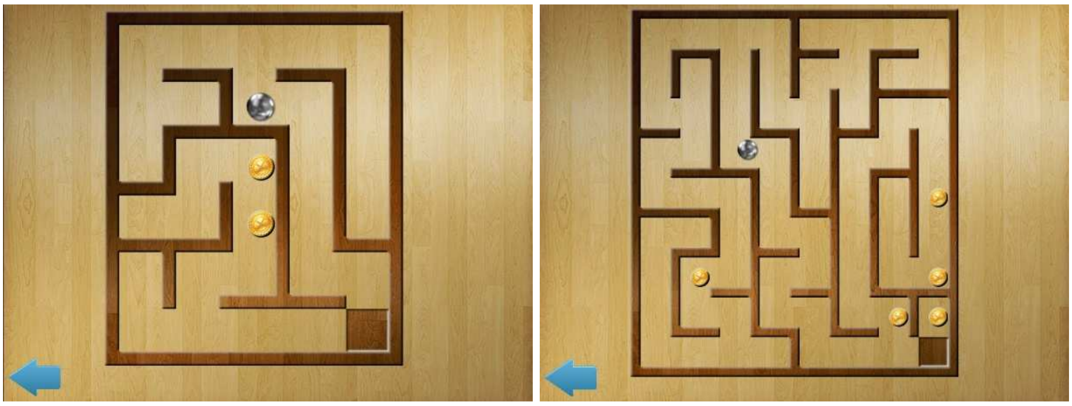


Figure 120: Labyrinth game (screenshots of different game levels)

As Figure 121 illustrates, there are two versions of the Labyrinth game which employ different interaction modalities. In the first one, the children try to tilt (on two axes) the Beantable's surface to guide the virtual steel ball to the end of the maze. Similarly, in the other version, the children try to tilt (on two axes) the augmented chair's sitting area with their body.

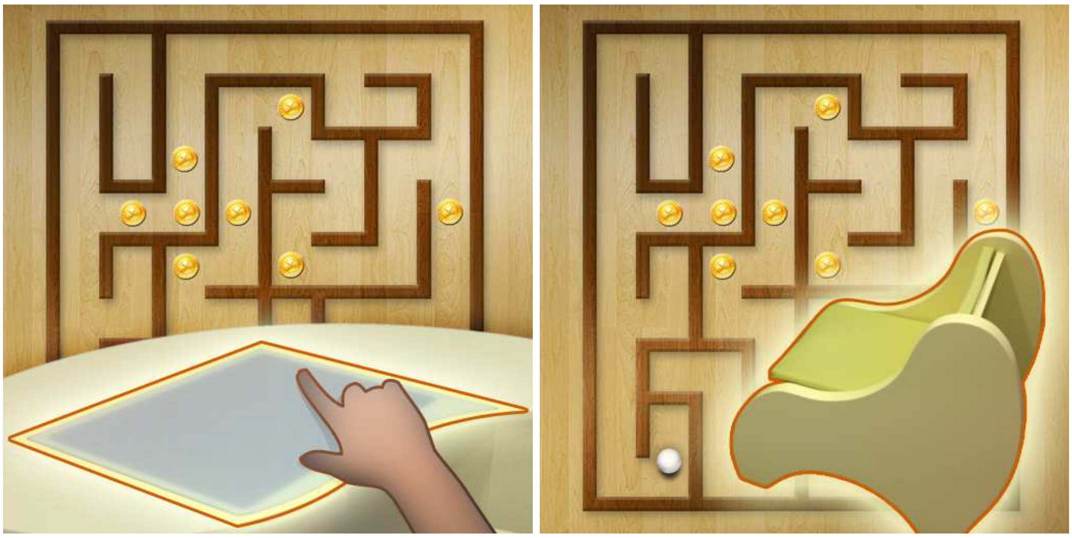


Figure 121: Labyrinth game's interaction modalities.

The Labyrinth game uses the Bean SDK's modules **SmartChairForcePressureRecognition** and **SurfaceForcePressureRecognition** as presented in 4.9.3. Therefore, the average force pressure distribution info is used to extract tilt information. Additionally, the more exerted pressure is applied, the more the virtual maze tilts, which results in a faster accelerated movement of the virtual ball.

6 Evaluation

6.1 Evaluation Methodology

For the purposes of this work and due to the diversity of the target user population, a mix of usability testing methods has been applied in a complementary way, including observation, which is the most common evaluation method, and picture cards method which can be used in usability testing with young children.

6.1.1 Observation method

Observation is an evaluation method that collects data by observing users' experiences with a product [128]. Often, observation-based usability evaluations are performed to determine quantitative measures like efficiency, effectiveness, etc. [100]. This method has been proven by countless research as one of the best techniques that could be implemented in usability testing. The approach is to have a session with end users, and observe them while performing a task.

A video camera captures what is happening to users, their interaction, and also the comments they made when interacting with the product. Additional video cameras can be used to record the users' facial expressions, body language, or any other important part of the user's environment.

Upon completion of each test, adult users (e.g., UG2, UG3) will need to answer post interview like **questionnaire** about other subjective inputs regarding the product such as ease of use and levels of satisfaction. Regarding children, the pictures card method can be used to help them express their thoughts.

6.1.2 The pictures card method

According to [19], the pictures card method is a new evaluation method that encourages young children to express both usability and fun issues while using a system. This method combines the traditional thinking-aloud method with picture cards that children can put in a box to indicate that there is a problem. An experiment to assess this method shows that children indicate more problems in a direct way with the picture cards than without the picture cards.

As mentioned in [19], there are several reasons why the picture cards method helps children to express more problems explicitly than when the facilitator just asks the child to verbalize their opinions:

- During the introduction the facilitator can use the picture cards to explain not only verbally but also visually what kind of information he/she is interested in. This combination of auditory and visual information adheres to the principles of multiple resources and redundancy gain and may make it easier for children to understand the explanation,

- During the test, the picture cards serve as memory aids for the things the evaluator is interested in, thereby putting "knowledge in the world" and relying less on long-term memory,
- Some children are able to verbalize what they think or feel, while others may be less verbally capable. With the picture cards method, less verbally capable children can express themselves explicitly without having to verbalize. This is a similar approach as interviewing techniques for young children.

A set of picture cards can be used in order to identify the main usability problems when users are children. For instance, the following usability problems have been defined in the literature concerning games for children [19]:

- **Usability problems at the cognitive level:** When a child encounters a usability problem on the cognitive level, he/she may not understand what to do, or what has happened,
- **Usability problems at the physical level:** When a child encounters a usability problem on the physical level, he/she may find it difficult.
- **Inefficiencies:** When a child encounters an inefficiency, he/she may think the system takes too long,
- **Challenge problems:** When a child experiences a problem related to a too high challenge level, he/she may find it too difficult to complete the task at hand.
- **Curiosity problems:** When a child experiences a curiosity problem, he/she may find the game boring,
- **Control problems:** When a child experiences a control problem, he/she may think the game takes too long.

Above each compartment the associated word was also visible. Children would get an explanation of the picture and the kind of situation they could use it for and could always ask for this explanation if they happened to forget it. An example set of pictures for the picture cards method appear in Figure 122.

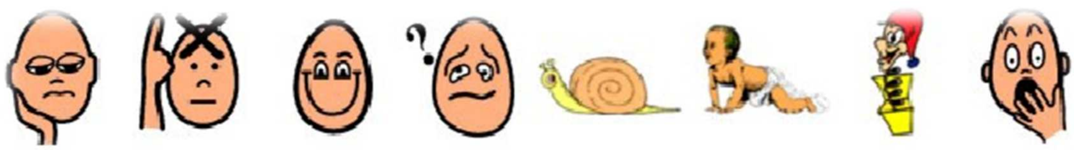


Figure 122: Picture Cards: From left to right, the used pictures for Boring, Don't know/understand, Fun, Difficult, This takes too long, Childish, Silly and Scary, respectively

6.2 Preliminary evaluation of the Beantable prototype

Two levels of evaluation have been conducted so far on the Beantable prototype. The first evaluation phase involved expert walkthroughs which were conducted by three user experience experts from the FORTH Human-Computer Interaction Lab. The second evaluation phase involved prototype testing with three children of ages 3, 3.5, and 6. The main objective was to assess the overall system usability and provide recommendations on how to improve design. The findings of both evaluations are reported below.

6.2.1 Expert walkthroughs evaluation

The experts were asked to play the preliminary developed smart games in order to uncover any potential violations of usability principles in the design, as well as identify any areas of the design that could potentially cause problems specifically to children.

Overall, the experts found the design of the game applications intuitive and engaging and pointed out only minor problems with the presentation of the menu and game information. Regarding the physical design of the table, the experts found it ergonomic and commented on how its circular design would facilitate cooperative gaming. Their only concern was that the chair was too heavy for a younger child to drag or lift to get closer to the table. The main suggestions that the experts made on the design and logic of the games were:

- Increasing the difficulty level of the memory card game (i.e., Pick & Match) by adding more cards every time the child solved the puzzle,
- Projecting a border on the screen the size of the actual puzzle (i.e., Playing jigsaw puzzles). The border image will help the child understand the physical dimensions of the puzzle and where to place the actual puzzle pieces,
- Projecting the image of the solved puzzle upon which the child could build the actual puzzle. This feature could be used for younger children that need help with solving the puzzle.

Changes based on the above observations and suggestions were implemented before moving to the second phase of the evaluation, the informal user-based evaluation.

Figure 123: From left to right: a girl 3.5 old and two boys 3.5 and 6 years old respectively

6.2.2 Informal user-based evaluation

Three children were invited to participate in the preliminary user-based evaluation of the Beantable prototype with the consent of their parents, who were present but did not play any specific role in the experiment. Two of the children were boys with ages 3.5 and 6, and one was a girl 3.5 years old. They were each asked by the evaluator to sit on the table and select a game from the menu to play. All three of them were able to open, close and play the games with very little given instructions by the evaluator. They all reached the highest level of difficulty of the memory card game (24 cards) and completed the two puzzles with ease (i.e., *Winnie* and *Three Little Pigs*). What was really impressive was the fact that even though the button labels that appeared in the dialog boxes were written in English, they knew which button to select based on its color, green for yes, red for no. They all expressed that they liked the games and that was also evident by the fact that they remained engaged throughout the evaluation and selected to play all three games.

6.3 User based evaluation

The final user based evaluation of the developed framework was conducted in order to explore usability, playability and applicability of the smart games implemented through the framework, as well as the usability and applicability of the provided monitoring tools in the context of the goals set by this work. Fourteen children, their parents, two occupational therapists, a psychologist and a special education teacher participated in the evaluation. Each child played the following smart games: a) The farm game, b) Tower game, and c) Mimesis game. At the same time, the child's parent(s) and the early intervention professionals were observing. An observation room for parents and experts was set-up in a remote location from the space where the actual evaluation was conducted. In this observation space, a projector was projecting live video from the evaluation space, while a personal computer was showing information regarding the current play performance achieved by the child. The children were encouraged to play freely without any external interventions by adults as shown in Figure 124.

Figure 124: Young children participating in the evaluation

After each evaluation session, children, parents and early intervention professionals were required to fill in a posttest questionnaire developed separately for each user group. The experts completed their questionnaire after the completion of all the evaluation sessions. Table 16 presents some indicative questions included in the questionnaires together with the ranking method employed (for more see APPENDIX VI - Questionnaires). For creating the questionnaires, the heuristics for evaluating playability were employed. These constitute a comprehensive set of heuristics for playability specifically tailored to evaluate video, computer and board games [49]. The interaction of children with the game was recorded in consensus with their parents, so as to allow posttest evaluation of their interaction with the system.

Children's Questionnaire sample (☹ ☹ 😐 ☺ ☺)

OVERALL
- How much did you like the game?
- Was the game too short, too long or just fine?

SYUSE
- How much different was this game from the traditional one?
- How often will you play this game?
- Did you have enough time between turns?

INFOQUAL
- How hard was for you to understand how to play?
- Did you know all the depicted animals?

INTERQUAL
- Are there enough options that you can do when playing?
- The size of the playing board and the pieces was too large, too small or fine for you?

Parents questionnaire sample (Yes/No, 1 to 10 and free text)

OVERALL
- Was the playing time satisfactory for the player?
- Was the game amusing and entertaining for your child?

SYUSE
- How much encouraging/ attractive for your child you find the presence or interaction with Max?
- Was there any safety issues?

INFOQUAL
- Did you find the game suitable for your kid's age?
- Was it easy for the child to use/interact alone/without the presence of an adult with the Beantable?

INTERQUAL
- Did you like the use of physical toys for playing the game?
- How much did you like the graphics/interface?

Experts Questionnaire sample (Yes/No, 1 to 10 and free text)

OVERALL
- How enjoyable was the game to replay?
- Was the theme proper for the age range is proposed for?
- Do you like Bean Administration Suite?

SYUSE
- Were the goals of the game clearly presented to the child?
- Was the game gradually increasing player's abilities?
- Was the application of Bean Administration Suite easy to use?

INFOQUAL
- Does the game present enough short – term goals throughout the game for ensuring entertainment / fun?
- Was player's failures resulted into positive feedback?
- Do you believe that this Bean Administration Suite is useful / necessary?

INTERQUAL
- How much immediate was the feedback received?
- Did the game provide enough information to get started?

Table 16: Indicative questionnaires for each group

6.3.1 Results

The evaluation results were extracted through an analysis of the recorded sessions and the results of the questionnaires. The recorded sessions were analyzed by usability experts to produce recommendations regarding further improvements of each game, while the questionnaires were used to calculate quality factors. Overall, four factors were calculated. The OVERALL factor expresses the overall satisfaction of the users regarding the system (calculated by the average of all the answers' grades). The SYUSE factor measures the satisfaction of users when using the system, while the INFOQUAL measures the information quality provided by the system. Finally, INTERQUAL is a factor that captures user satisfaction regarding the interface provided by the system. The linking of questions to the calculated usability factors is also presented in Table 16.

Figure 125 represents these factors for children. The **OVERALL** factor shows that children were generally satisfied (~75% of children scored 4 and 5 to all questions) by the overall usability of the system. However, there are 20,45% of the children who state that they were not fully satisfied. The **SYUSE** factor shows that children were generally satisfied (~85% of children scored 4 and 5) by the overall satisfaction by using the system. However ~15% of the users state that they were little or not satisfied. Regarding the quality of information (**INFOQUAL**) ~76% of the users scored 4 or 5. However, there are ~24% of children that scored 1 to 3 which implies that are is a substantial amount of users that requires some form of improvement in the way that information is presented. Regarding the interaction quality (**INTERQUAL**) ~64% of the users scored 4 and 5. However, there are ~22% of the users that scored 3 and ~13% that score 1 and 2 which implies that is a substantial amount of users that requires some form of improvement in their interaction with the games.

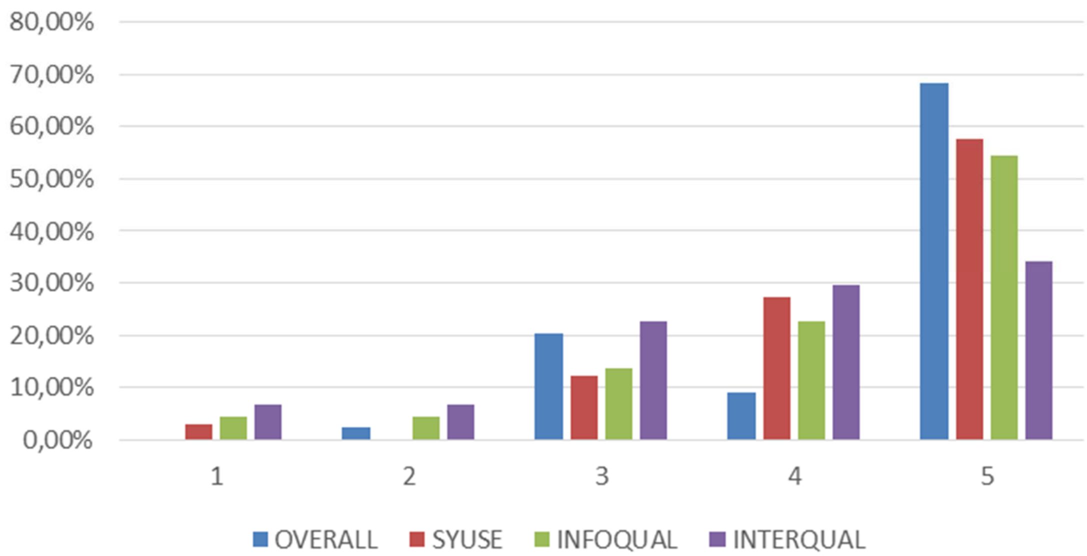


Figure 125: Usability factors (children)

Regarding playability, children reported a few problems for game play found in the aforementioned 3 smart games. Children commented that sometimes Max was unable to understand their answers. This indicates that the speech recognition module of Nibbler has weakness to successfully recognize the child's speech. This inability reveals the limitation set by the fixed vocabulary used in speech recognition module which has to be set in advance. As a result, synonyms of an expected answer (e.g., "yes" has synonyms like *sure*, *certainly*, *indeed*, etc.) or its equivalent expressions (e.g., *of course*) could not recognized. Additionally, at the time of framework implementation, there was no official support for speech recognition in Greek language. To overcome this limitation, the selected vocabulary contains a number of Greek words written in Latin alphabet. Indubitably, this method reduces the speech recognition efficiency. Regarding farm game, the majority of the children commented that the fourth level was not quite similar to the previous and they seemed to be surprised. On the other hand, some children noticed that tower game, as well as the mimesis game remained the same (i.e., looked similar) across different levels.

Furthermore, some children found it inconvenient to use the Beantable's multitouch interactive surface. For instance, the tower game did not succeed to entirely satisfy them due to the fact that the system recognized all touch events, even those that occurred with no intention. More specific, some times, when another object accidentally touches the surface of the interactive area (e.g., the sleeve of the young child while playing), the system may automatically recognized it as interaction input that has been done from the user in order to continue playing with the game. As a result, the tower game interprets as mistake the child's path selection because it contains incorrent tiles. Regarding Mimesis game, some children were unable to keep an optimal distance from the large display. In most cases, children went to stand just in front of the TV. This had as a result the inability of Nibbler's skeleton tracking module to recognize their appearance due to the limitations set by the sensor device (i.e., Kinect sensor has a practical ranging limit of 1.2–3.5m).

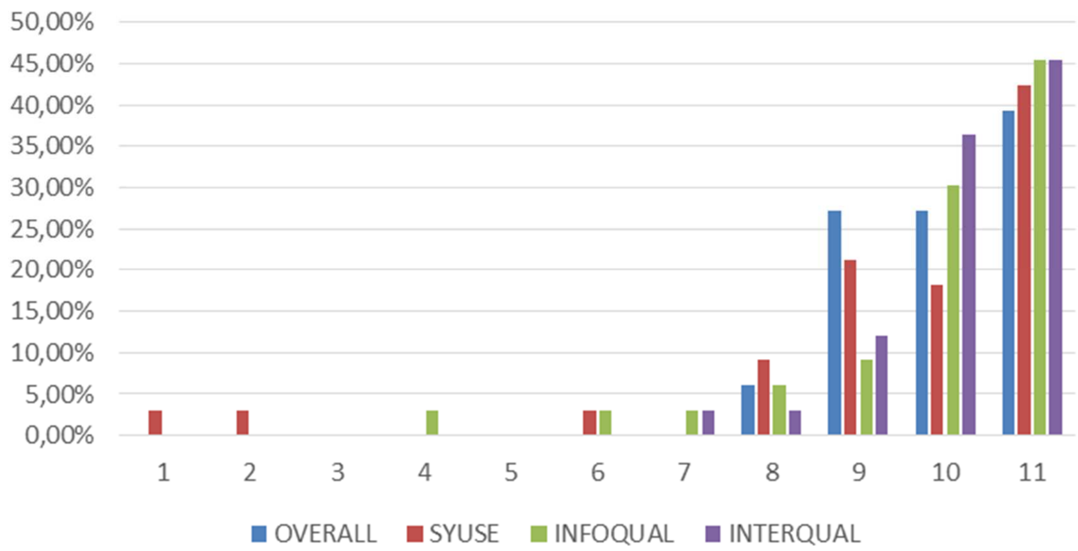


Figure 126: Combined chart of usability factors (parents)

In the case of parents, Figure 126 shows that the game scored well in all the calculated usability factors with scores for 8 to 10, gathering the majority of their goals for all usability factors.

Experts rating was also very positive (see Figure 127) and this was also expressed during the informal interviews conducted after the completion of the evaluation. The experts commented very positively both the design of the games and the adaptation mechanism used to dynamically adapt the games to the developmental characteristics of children while playing.

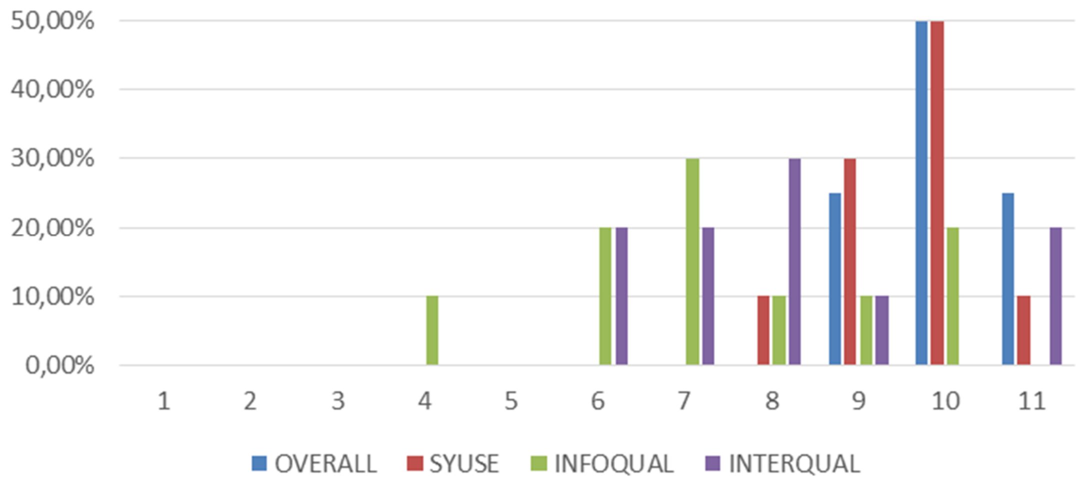


Figure 127: Combined chart of usability factors (experts)

6.3.2 Parents' and early interventions professional's comments

One important goal of the evaluation was also to gather comments from users so as to help the improvement of the games and the framework itself. The collection of these comments was done by discussing with them in the context of informal interviews after the completion of the evaluation. For the farm game, some **parents** (mostly of older children) commented that the game should give feedback every time the child places a physical object at its correct virtual position. This could be a visual or an audio feedback to give the feeling that the item is correctly placed as happening in the actual traditional game. Moreover, parents think that the voice of Max should be louder and less computerized. **Early intervention professionals** aptly commented that Max did not introduce comprehensively the rules of each smart game. In detail, each smart game should always demonstrate the "how to play" guidelines only for the first time the child chose to play. Furthermore, early intervention professionals noticed that on the one hand Max was very slow in his reactions and additionally that in some games idle time was expiring too early without allowing children to think or plan how to play. They also expressed their concern about the size of the secondary screen presenting Max that in their opinion should be bigger. Both parents and early intervention professionals commented that the smart games could successfully provide useful information about the children's skills and abilities with only one small exception. When a young child started playing the farm game, he was very quiet and did not seem to react at the request of the Max to place his identification card. Unfortunately, he appears to have some problem of communication, something that the system was not able to monitor because of the uncompleted identification procedure. On the other hand, the system was successfully extracted indications of possible skill immaturities for a child of 5.5 years old known to face learning difficulties.

7 Discussion and Future work

7.1 Discussion

The work presented in this thesis aims to build an Ambient Intelligence (AmI) environment for supporting young children through playing. To this end, an underlying AmI technological framework (*Bean framework*) has been described which offers automatic and proactive services. In detail, the proposed AmI technological framework supports the design, development and deployment of smart games with innovative characteristics. When these games are deployed in an AmI powered environment, they become capable of monitoring and evaluating children's skills and abilities, while, on the other hand, enhancing children playing experience as they adapt to meet their continuously changing playing maturity. At the same time, children needs for activity, exercise and pleasure are optimally covered while also providing opportunities for creativity. As a result, the AmI environment provides functionalities and intuitive tools in order to monitor, evaluate and enhance various children abilities such as thinking, problem solving, etc. By taking into account the child's individual preferences, habits and abilities, the AmI environment is able to adapt the interaction model, and therefore to enhance the child's playing experience.

The work presented in this thesis includes the design and implementation of a collection of augmented artifacts including an augmented interactive table suitable for young children (i.e., Beantable), an augmented chair, a digital pen, various ready to use ICT devices such as an iPad, augmented digital dice, etc. The implementation of these artifacts was conducted taking into account the basic principles of AmI and ubiquitous computing making devices unobtrusive and technology hidden or embedded in traditional everyday objects and furniture. At the same time, general health and safety requirements for children were also taken into account.

Furthermore, novel interaction metaphors and techniques and the appropriate administration and training facilities were supported by the development of a unified interaction framework called Nibbler, which offers among others: a) gesture/posture recognition, b) head pose estimation, c) face tracking, d) skeleton tracking, e) speech recognition, etc.

Transformation of physical artifacts that typically possess no technology into augmented physical objects such as puzzle pieces, identity cards, toys, etc., has been achieved by employing computer vision technologies for visual markers recognition.

ADAM is an adaptation infrastructure mechanism for automated extraction of knowledge, based on interaction monitoring, to offer indications regarding the children development state, maturity level and skills.

Additionally, this thesis presented the design and implementation of a novel, remotely controlled three-dimensional full body avatar simulation framework, targeted to edutainment and instructor-student interaction. The main innovation focuses on multi-presence gamified educational scenarios in multiple desktop computers and mobile devices.

Also, the work presented in this thesis includes the design and implementation of a new scripting language, called ACTA, for facilitating activity analysis during the smart game design process by early intervention professionals. Additionally, developers can use ACTA not only for developing event-driven sequential logic games, but also for applications of behavior composed of a finite number of states, transitions between those states, and actions as well as for application based on rules driven workflows.

Using the proposed framework, popular games were designed with the assistance of HCI experts and occupational therapists and used as smart games (e.g., puzzles, card games, labyrinths, the tower game, the farm game, the mimesis game) acting as the required software layer to host the novel interaction channels proposed by this research work.

Likewise, the work conducted in this thesis includes the implementation of a child monitoring framework based on occupational therapy's expertise aiming at early detection of children's potential delays to be further investigated and diagnosed if necessary. To this end, content editing and interaction monitoring tools (i.e., Bean Administration Suite) were implemented to support the needs of parents and therapists based on the extracted requirements. For example, parents are provided with general information (in a pleasant and aesthetic way) about their child's physical and mental development progress, as well as indications of a possible skill immaturity. Similarly, early intervention professionals are provided with extensive data in addition to the full interaction history for reasoning about whether the child is meeting all the necessary developmental milestones.

The following publications stem from the work presented in this thesis:

- Zidianakis, E., Antona, M., Paparoulis, G., & Stephanidis, C. (2012). An augmented interactive table supporting preschool children development through playing. In the Proceedings of the 2012 AHFE International Conference (4th International Conference on Applied Human Factors and Ergonomics), San Francisco, California, USA, 21-25 July (pp. 744-753).
- Zidianakis, E., Partarakis, N., Antona, M., & Stephanidis, C. (2014). Building a Sensory Infrastructure to Support Interaction and Monitoring in Ambient Intelligence Environments. In Distributed, Ambient, and Pervasive Interactions (pp. 519-529). Springer International Publishing.
- Zidianakis, E., Papagiannakis, G., & Stephanidis, C. (2014). A cross-platform, remotely-controlled mobile avatar simulation framework for AmI environments. In SIGGRAPH Asia 2014 Mobile Graphics and Interactive Applications (p. 12). ACM.

The developed augmented interactive table (i.e., Beantable) was presented in the exhibition of the HCI International Conference 2014 (HCII). HCII is an ideal opportunity to exhibit products and services to an international audience of about 2,000 researchers, academics, professionals and users in the field of HCI. Beantable received plenty of enthusiastic comments, as well as many potential purchasers have shown interested.

The newspaper “Kathimerini” featured an article on the Greek achievements in the domain of Ambient Intelligence (see APPENDIX VII – Article in the newspaper “Kathimerini”). In detail, the article expresses enthousiastic comments on the ergonomic design of the Beantable, the employed sensory infrastructure, as well as the innovative features that smart games provide.

The aforementioned framework was deployed within a simulation space in the FORTH’s AmI Facility and a two-phase evaluation with children, parents and early intervention professionals was conducted in order to identify potential usability barriers prior to the practical exploitation of the concepts. The first phase of the evaluation was conducted early after implementing the basic technological infrastructure, so as to primarily measure user acceptance as well as the ergonomic design of the settled artifacts. Participants of this evaluation were children aged from three to six years old. The second phase of the evaluation was conducted after the complete deployment of the framework, in order to validate its value for each target group. Therefore, the evaluation user base was expanded, including young children in the age range of three to seven years old, their parents and early intervention professionals.

7.2 Future work

Two main directions of further work are anticipated in a path towards supporting the fruition of in Ambient Intelligence Environments. The first direction concerns the improvement of developed infrastructure and applications and their evolution from an in-vitro prototype to a mature software product. Towards this end, a number of planned improvements of the application are stemming from the conducted evaluations. To this end, there is a number of less critical usability errors identified using the user based evaluation of the applications that should be addressed. These results are important as mirroring issues of the end-users experience with the system.

Regarding future improvements of Nibbler two are the main directions. The first planned improvement is the integration of gaze tracking functionality so as to complete the suite of head tracking facilities. The second planned improvement is the integration of finger tracking functionality so as to expand the monitoring capabilities of the framework, e.g., hand-eye coordination. Physical object recognition may include the recognition of objects based on their entirely visual characteristics in three-dimensional space without adding special visual markers on them. Regarding future improvements of ADAM, more types of adaptation suggestions will be introduced in order to facilitate automatic UI adaptation and the efficient selection of available modalities based on children’s skills and abilities that continuously change. Further gameplay and physiological data will be collected and analyzed in order to enhance the adaptation infrastructure mechanism with more accurate multi-modal predictors of game experience.

Regarding the avatar framework, it is considered crucial to extend the support for more platforms such as Android phones to holistically address the mobile computing market. Additionally, an evaluation strategy will be designed with the active contribution of the end users and research questions will be formulated around user experiences. The ACTA IDE will be enhanced with an event driven visualization mechanism. Furthermore, ACTA capabilities will expand so as authors be able to set alternative sequential finite states based on some condition. For example, in case that the macros defined within a state are incompatible according to child's profile, the state can be automatically replaced by others more suitable. Additionally, the next step is to integrate together both ADAM and ACTA scripting language in order to further minimize the required time for a smart game to be designed and developed. Finally, an evaluation will be designed and conducted to measure ACTA's applicability and usability from end-users perspective.

References

1. Aarts E., Encarnacao J.L. (2008). True Visions. The Emergence of Ambient Intelligence. Springer, ISBN 978 ‐ 3 ‐ 540 ‐ 28972 ‐ 2.
2. Aarts, E. H., Marzano, S. (2003). The new everyday: Views on ambient intelligence. 010 Publishers, Rotterdam
3. Aarts, E.H.L., Harwig, H., Schuurmans, M. (2001), Ambient Intelligence, in: J. Denning (ed.) The Invisible Future, McGraw Hill, New York, NY, USA, pp. 235-250.
4. Adamson, L. B., Bakeman, R., (1999). "Viewing variations in language development: The Communication Play Protocol." Augmentative and Alternative Communication (Newsletter for ASHA Division 12) 8
5. Ager, R., Kendall, M. (2003). Getting it right from the start: A case study of the development of a foundation stage learning and ICT strategy in Northamptonshire, UK.
6. Ailisto, H., Kotila, A., StrÃmmer, E. (2003). "Ubicom applications and technologies". Presentation slides from ITEA 2003, Oulu, Finland
7. Akiyama Yasushi, Sageev (2008). A Multimodal Input Device for Music Authoring for Children. In the proceeding of 14th International Multimedia Modeling Conference, MMM Kyoto, Japan
8. Alcañiz, M., Rey, B. (2005). New Technologies For Ambient Intelligence, IOS Press, http://www.ambientintelligence.org
9. Altman, D.G. (1991). Some common problems in medical research. Practical statistics for medical research. New York, N.Y; Chapman and Hall, pp. :396-438
10. Antona, M., Margetis, G., Ntoa, S., Leonidis, A., Korozi, M., Paparoulis, G. & Stephanidis, C. (2010). Ambient Intelligence in the classroom: an augmented school desk. In W. Karwowski & G. Salvendy (Eds.), Proceedings of the 2010 AHFE International Conference (3rd International Conference on Applied Human Factors and Ergonomics), Miami, Florida, USA, 17-20 July. [CD-ROM]. USA Publishing (ISBN: 978-0-9796435-4-5).
11. Antona, M., Savidis, A., Stephanidis, C. (2005). A support tool for the design of user interface adaptation. In C. Stephanidis (Ed.) Universal Access in HCI: Exploring New Interaction Environments. Volume 7 of the Proceedings of the 11th International Conference on Human-Computer Interaction (HCI International 2005), 22-27 July 2005, Las Vegas, Nevada, USA.
12. Arita, D., Taniguchi, R.I. (2004). Real-time human proxy: An avatar-based interaction system. In Knowledge-Based Intelligent Information and Engineering Systems (pp. 419-425). Springer Berlin Heidelberg
13. Ayers, A.J., (1980). Southern California Sensory Integration Test (SIPT) Manual. WPS (1985)
14. Azuma, R. T. (1997). A survey of augmented reality. Presence, 6(4), pp. 355-385.
15. Bajura, M., Fuchs, H., Ohbuchi, R. (1992). Merging Virtual Objects with the Real World: Seeing Ultrasound Imagery Within the Patient. In Proceedings of SIGGRAPH '92, New York: ACM Press, pp. 203-210.
16. Baldassarri S., Cerezo E., Serón F., (2008). Maxine: A platform for embodied animated agents. Computer & Graphics. ISSN: 0097-8493. Vol. 32 (4), pp.430-437. August 2008.
17. Ball, R. S. (1977). The Gesell developmental schedules: Arnold Gesell (1880–1961). Journal of abnormal child psychology, 5(3), pp. 233-239.
18. Baranek GT, Barnett C, Adams E, Wolcott N, Watson L, Crais E (2005) Object play in infants with autism: methodological issues in retrospective video analysis. Am J Occup Ther 59(1):20–30

19. Barendregt, W., Bekker, M. M., (2008). Development and Evaluation of the Picture Cards Method, Cognition Technology and Work. Published by the British Computer Society Volume 2 Proceedings of the 21st BCS HCI Group Conference HCI 2007
20. Begg, C.B. (1987). Biases in the assessment of diagnostic tests. Stat Med pp.411-423
21. Bencina, R., Kaltenbrunner, M. (2005). The Design and Evolution of Fiducials for the reacTIVision System, Proceedings of the Third International Conference on Generative Systems in the Electronic Arts (Third Iteration), Melbourne, pp. 97-106
22. Bergen, D. (1998). Stages of Play Development. In D. Bergen (Ed.), Play as a medium for learning and development (pp.71–93). Olney, MD: Association for Childhood Education International
23. Berger-Vachon, C. (2006). Virtual reality and disability. Technology and Disability, 18, 163-165.
24. Bernardes, J., Tori, R., Nakamura, R., Calife, D., Tomoyose, A. (2008). Augmented reality games. Extending Experiences: Structure, analysis and design of computer game player experience, 1, 228-246.
25. Blasco, P. A. (1991). "Pitfalls in Developmental Diagnosis." Pediatric Clinics of North America 38 14251438.
26. Blended Reality report. Technology Horizons Program (2009). Institute for the Future, Palo Alto: http://www.iftf.org/uploads/media/SR-122~2.PDF
27. Bohn J. (2004) The Smart Jigsaw Puzzle Assistant: Using RFID Technology for Building Augmented Real-World Games; Institute for Pervasive Computing; Zurich, Switzerland
28. Brenner D., Merkel C., (2009). Autonomous Weapon Turret Independent Investigation: Target Tracking Using Intel's OpenCV Library
29. Bricken, M., Byrne, C. M. (1993). Summer students in virtual reality. Virtual reality: Applications and exploration, 199-218.
30. Bricker, D. D., Squires, J., Potter, L. W., & Twombly, R. E. (1999). Ages & Stages Questionnaires (ASQ): A Parent-Completed, Child-Monitoring System. Paul H. Brookes Publishing CO.
31. Bulman, J., Crabtree, B., Gower, A., Oldroyd, A., Lawson, M., Sutton, J. (2004) Mixed reality applications in urban environments. BT Technology Journal 22, 3 pp. 84-94.
32. Bunt, H., Beun, R. - J., Borghuis, T. (1995) Proc. of the International Conference on Cooperative Multimodal Communication CMC/95, 24-26 May 1995, Eindhoven. Samenwerkingsorgaan Brabantse Universiteiten (SOBU).
33. C. May, L. Gask, T. Atkinson, N. Ellis, F. Mair, A. Esmail, (2001). Resisting and promoting new technologies in clinical practice: the case of telepsychiatry, Soc. Sci. Med. 52 (12) pp. 1889–1901. •
34. Camp, B.W. (2007) Evaluating bias in validity studies of developmental/ behavioral screening tests, pp. 234-240.
35. Cason, J., Jacobs, K., (2014). Snapshots of current telehealth applications in occupational therapy. The american occupational therapy association. volume 19, issue 14
36. Cavazza. M., Aylett, R., Dautenhahn, K., Fencott, C. F. Charles. (200) Interactive Storytelling in Virtual Environments: Building the "Holodeck", Proceedings of VSMM, pp. 678- 687.
37. Chaczko, Z., Yeoh, L. A., Mahadevan, V. (2010). A Preliminary Investigation on Computer Vision for Telemedicine Systems using OpenCV. InMachine Learning and Computing (ICMLC), 2010 Second International Conference on (pp. 42-46). IEEE.

38. Cheng J., Zhou B., Sundholm M., Lukowicz P. (2013) Smart Chair: What Can Simpe Pressure Sensors under the Chairs' Legs Tell Us about User Activity?; German Research Center for Artificial Intelligence; Germany; UBICOMM 2013, The Seventh International Conference on Mobile Ubiquitous Computing Systems, Service and Technologies
39. Chignell, M. H., Hancock, P. A. (1998) Intelligent Interface Design. In: M. Helander (ed.), Handbook of Human-Computer Interaction. Amsterdam: North-Holland, pp. 969-995.
40. Clarke, A. M. (2006). Young Children and ICT-current issues in the provision of ICT technologies and services for young children. ETSI White Paper, (2).
41. Consolidated financial highlights. Kyoto (Japan): Nintendo; 2010. Available from: http://www.nintendo.co.jp/ir/pdf/2010/101028e.pdf
42. Cook. D. J., Das S. K. (2007). How smart are our environments? An updated look at the state of the art; Pervasive and Mobile Computing, Design and Use of Smart Environments
43. Cooper, B., Brna, P. (2002). Hidden curriculum, hidden feelings: Emotions, relationships and learning with ICT and the whole child. Paper presented at the BERA Conference, Exeter, September, 2002.
44. Costello, J., Ali, F. (1971). Reliability and validity of Peabody Picture Vocabulary Test scores of disadvantaged preschool children. Psychological Reports, 28(3), 755-760.
45. Cruz-Neira, C., Sandin, D. J., Defanti, T. A., Kentyon, R. V., Hart, J. C. (1992). The CAVE: Audio Visual Experience Automatic Virtual Environment. Communications of the ACM, Vol. 35 (6), pp. 65.
46. Czyzewski A., Odya P., Grabkowska A., Grabkowski M., Kostek B.; (2010). Smart Pen – New multimodal computer control tool for dyslexia therapy; Gdansk University of Technology, Multimedia Systems Department; Poland
47. Dede, C. (2005). Planning for neomillennial learning styles. Educause Quarterly 28, 1
48. Delapierre, G., Grange, H., Chambaz, B., Destannes, L. (1983). Polymer-based capacitive humidity sensor, Sensors and Actuators 4 (1) pp. 97-104.
49. Desurvire, H., Caplan, M., Toth, J. A. (2004). Using heuristics to evaluate the playability of games. In CHI'04 extended abstracts on Human factors in computing systems 1509-1512. ACM.
50. Domdouzis, K., Kumar, B., Anumba, C. (2007). Radio-Frequency Identification (RFID) applications: A brief introduction. Advanced Engineering Informatics, 21(4), 350-355.
51. Doulgeraki, C., Partarakis, N., Mourouzis, A, Stephanidis, C. (2008). A Development Toolkit for Unified Web-Based User Interfaces. Computers Helping People with Special Needs, 2008, pp 346-353
52. Druzhkov, P. N., Erukhimov, V. L., Zolotykh, N. Y., Kozinov, E. A., Kustikova, V. D., Meerov, I. B., Polovinkin, A. N. (2011). New object detection features in the OpenCV library. Pattern Recognition and Image Analysis, 21(3), 384-386.
53. Ducatel K., Bogdanowicz M., Scapolo F., Leijten J.,Burgelman J-C., (2001). Scenarios for Ambient Intelligence in 2010, IPTS-Seville
54. Dunston, P. S., Wang, X. (2005). Mixed reality based visualization interfaces for architecture, engineering, and construction industry. Journal of Construction Engineering and Management 131, 12, pp. 1301-1309.
55. E. C. I. S. Directorate-General. (2001). ISTAG report: Scenarios for ambient intelligence in 2010.
56. Egges, A., Papagiannakis, G., Magnenat-Thalmann, N. (2007). Presence and interaction in mixed reality environments. In The Visual Computer, 23(5), 317-333.

57. Eisenmann, J. C., Heelan, K. A., & Welk, G. J. (2004). Assessing body composition among 3 to 8 year old children: Anthropometry, bia, and dxa. Obesity Research, 12(10), 1633-1640.
58. Erickson, T.D. (1990). Working with Interface Metaphors. In B. Laurel (Ed.), The Art of Human-Computer Interface Design (pp. 65-73). Reading, MA: Addison-Wesley Publishing Company, Inc
59. Ermes, M., Parkka, J., Mantyjarvi, J., Korhonen, I. (2008) Detection of daily activities and sports with wearable sensors in controlled and uncontrolled contitions, IEEE Transactions on Information Technology in Biomedicine 12 pp. 20-26.
60. Eyles, M., & Eglin, R. (2008). Ambient games, revealing a route to a world where work is play? International Journal of Computer Games Technology, 7.
61. Farringdon, J., Nashold, S., (2005). Continuous Body Monitoring. Ambient Intelligence for Scientific Discovery, pp 202-223.
62. Feiner, S., MacIntyre, B., Seligmann, D. (1993) Knowledge-Based Augmented Reality. Communications of the ACM, Vol. 36(7), pp. 53-62.
63. First, L., J. Palfrey. (1994). "The Infant or Young Child with Developmental Delay." The New England Journal of Medicine 330, no. 7 pp. 478-483.
64. Fisher, A. G. (2006). Overview of performance skills and client factors. In H. M. Pendleton, & W. Schultz-Krohn (Eds.), Pedretti's occupational therapy: Practice skills for physical dysfunction (6th ed., pp. 372-402). St. Louis MO: Mosby Elsevier.
65. Fisher, K., Hirsh-Pasek, K., Golinkoff, R. M., Singer, D. G., Berk, L. (2011). Playing around in school: Implications for learning and educational policy. In A. D. Pellegrini (Ed.), Oxford Handbook of the Development of Play. Oxford University Press.
66. Fitzmaurice, G. (1993). Situated information spaces: Spatially aware palmtop computers. Communications of the ACM, 36 (7), 38-49.
67. Fontijn, W, Mendels, P. (2005). StoryToy the interactive storytelling toy. In The Second International Workshop on Gaming Applications in Pervasive Computing Environments at Pervasive 2005.
68. Frankenburg, W. K., Dodds, J. B. (1967). The Denver developmental screening test. The Journal of pediatrics, 71(2), pp. 181-191.
69. Frankenburg, W. K., Dodds, J., Archer, P., Shapiro, H., Bresnick, B. (1992). The Denver II: a major revision and restandardization of the Denver Developmental Screening Test. Pediatrics, 89(1), pp. 91-97.
70. Frankenburg, W. K., Goldstein, A. D., & Camp, B. W. (1971). The revised Denver Developmental Screening Test: Its accuracy as a screening instrument. The journal of Pediatrics, 79(6), pp. 988-995
71. Frankenburg, W.K. (2002). "Developmental Surveillance and Screening of Infants and Young Children". Pediatrics 109 (109) pp. 144–145.
72. Frankenburg, W.K., Dodds, J., Archer, P. et al. (1990). The DENVER II Technical Manual 1990, Denver Developmental Materials, Denver, Co.
73. Franklin, M. J., Lee Giles, C. (2000). Self-adaptive user profiles for large-scale data delivery, In Proceedings of the 16th IEEE International Conference on Data Engineering (ICDE'00).
74. Freedman, T. (2001). Primary effects. The Guardian. http://education.guardian.co.uk/elearning/story/0,,789479,00.html. Retrieved April 26, 2004
75. Garzotto, F., Forfori, M., (2006). FaTe2: Storytelling edutainment experiences in 2D and 3D collaborative spaces. Proceedings of the 2006 conference on interaction design and children, Tampere, Finland, pp. 113-116.
76. Georgalis, Y., Grammenos, D., Stephanidis, C., (2009). Middleware for Ambient Intelligence Environments: Reviewing Requirements and Communication

Technologies. In: Stephanidis, C. (ed.) UAHCI 2009, Part II. LNCS, vol. 5615, pp. 168–177. Springer, Heidelberg

77. Glascoe, Frances Page et al. (1992). "Accuracy of the Denver-II in Developmental Screening". Pediatrics (89) pp. 1221–1225.
78. Goldman, D., (2012). "Google unveils 'Project Glass' virtual-reality glasses". Money (CNN)
79. Goldstein, J. (2012). Play in Children's Development, Health and Well-Being. Brussels. Toy Industries of Europe (TIE)
80. Goode, S., Diefendorf, M., & Colgan, S. (2011). The importance of early intervention for infants and toddlers with disabilities and their families. National Early Childhood Technical Assistance Center.
81. Grammenos, D., Margetis, G., Koutlemanis, P., & Zabulis, X. (2012). Paximadaki, the game: Creating an advergame for promoting traditional food products. In Proceeding of the 16th International Academic MindTrek Conference (pp. 287-290). ACM.
82. Grammenos, D., Zabulis, X., Michel, D., Padeleris, P., Sarmis, T., Georgalis, G., Koutlemanis, P., Tzevanidis, K., Argyros, A.A., Sifakis, M., Adam-Veleni, P., Stephanidis, C. (2012). Macedonia from Fragments to Pixels: A permanent exhibition of interactive systems at the Archaeological Museum of Thessaloniki. In Marinos Ioannides; Dieter Fritsch; Johanna Leissner; Rob Davies; Fabio Remondino & Rossella Caffo, ed., 'EuroMed' , Springer, , pp. 602-609. DOI: http://link.springer.com/chapter/10.1007/978-3-642-34234-9_62
83. Granlund, M., Eriksson, L. Ylven. R. (2004). Utility of the International Classification of Functioning, Disability and Health participation dimension in assigning ICF codes to items for extant rating instruments. J. Rehab. Medicine. 36(3) 130-137.
84. Hailpern, J. M., Karahalios, K., Halle, J., DeThorne, L., Coletto, M.-K. (2008). Visualizations: speech, language & autistic spectrum disorder. Extended Abstracts of Conference on Human Factors in Computing Systems (CHI '08) (pp. 3591-3596). Florence, Italy: ACM Press.
85. Halton J., (2008), Virtual Rehabilitation with Video Games: A new frontier for occupational therapy. Occupational Therapy Now, 9(6), 12-14.
86. Hayes, G. R., Gardere, L. M., Abowd, G. D., Truong, K. N. (2008). CareLog: a selective archiving tool for behavior management in schools. Conference on Human Factors in Computing Systems (CHI '08) (pp. 685-694). Florence, Italy: ACM Press.
87. Hayes, G. R., Truong, K. N., Abowd, G. D., Pering, T. (2005). Experience buffers: a socially appropriate, selective archiving tool for evidence-based care.
88. Herr, J. (2000). Creative learning activities for young children. Cengage Learning.
89. Hewes, J., (2006). “Let the Children Play:Nature's Answer to Early Learning”, PhD Chair of the Early Childhood Education Program, Grant MacEwan College, Alberta, Canada
90. Hinske, S., & Langheinrich, M. (2008). RFIDice–Augmenting Tabletop Dice with RFID. J. Herder (Ed.).
91. Hoff, E. (2013). Language development. Cengage Learning. pp. 23-24
92. Holden, G. W. (2010). Childrearing and Developmental Trajectories: Positive Pathways, Off‐Ramps, and Dynamic Processes. Child Development Perspectives, 4(3), pp. 197-204.
93. Holden, M. K. (2005). Virtual environments for motor rehabilitation: Review. Cyberpsychology and Behavior, 8, p.188
94. Hongpaisanwiwat, C., Lewis, M. (2003). Attentional effect of animated character. In Proceedings of the human-computer interaction, 423-430.
95. http://www.w3.org/TR/owl-ref/ (2004), OWL Web Ontology Language Reference. W3C Recommendation, 10 February 2004.

96. Hu, M.-K. (1962). Visual pattern recognition by moment invariants. IRE Transactions on Information Theory, Vol. 8, pp. 179-187.
97. Hub, A., Hartter, T., Ertl, T. (2006). Interactive tracking of movable objects for the blind on the basis of environment models and perception-oriented object recognition methods. In Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility (pp. 111-118). ACM.
98. Hurlock, Elizabeth B. (1978). Child growth and development. Tata McGraw-Hill Education
99. Hutinger, P., Rippey, R. Johanson, J. (1999). Findings of Research Study on Effectiveness of a Comprehensive Technology System Demonstrate Benefits for Children and Teachers, ACTTive Technology, Winter Issue. Retrieved 29 September, 2003, from http://www.wiu.edu/users/mimacp/wiu/articles.html
100. ISO 9241-11, (1998) Ergonomic requirements for office work with visual display terminals (VDTs) - Part 11: Guidance on usability.
101. IST Advisory Group, Ambient Intelligence: from Vision to Reality; For participation – in society & business. ISTAG Report Sep.' 2003
102. ISTAG, Ambient Intelligence: from Vision to Reality, in Ambient Intelligence, Lakatta Riva G, Vatalaro F, Davide F, Alcañiz M (Eds.), IOS Press, 2005.
103. J. P. Hourcade, B. Bederson, A. Druin G. Taxen. (2002). KidPad: collaborative storytelling for children. In CHI '02 Extended Abstracts on Human Factors in Computing Systems, Minneapolis, Minnesotta, USA, pp. 500-501.
104. Jackson, R. L., Fagan, E. (2000). Collaboration and learning within immersive virtual reality. Collaborative Virtual Environments 2000.
105. Jefferson College of Health Science. Information and Communication Technology in Health: a review of the Literature. Volume 3 No.1, 2008
106. Jia, P., Hu, H. H., Lu, T., Yuan, K. (2007). Head gesture recognition for hands-free control of an intelligent wheelchair. Industrial Robot: An International Journal, 34(1), 60-68.
107. Jones, E., Reynolds, G. (1992). The play's the thing: Teachers' roles in children's play, p. 19
108. Kaliouby, R. e., Robinson, P. (2005). The emotional hearing aid: an assistive tool for children with Asperger syndrome. Universal Access in the Information Society.
109. Kalliala, M. (2006). Play Culture in a changing world. Berkshire, England: Open University Press.
110. Kaufmann, H. (2003) Collaborative augmented reality in education. In position paper for keynote speech at Inagina 2003 conference.
111. Kidd C.D., Orr R., Abowd G.D., Atkeson C.G., Essa I.A., MacIntyre B., Mynatt E., Starner T.E., (1999). Newstetter W.: The Aware Home: A Living Laboratory for Ubiquitous Computing Research, in: Proc. Of the 2nd Intern. Workshop on Cooperative Buildings, Integrating Information, Organization, and Architecture (CoBuild'99), pp. 191-198, Springer Verlag, London, UK
112. Kientz J. K. Abowd G. D. (2009). KidCam: Toward an Effective Technology for the Capture of Children's Moments of Interest. University of Washington, School of Interactive Computing and Health Systems Institute. Georgia, USA
113. Kientz, J. A., (2008). Decision Support for Caregivers through Embedded Capture and Access. PhD thesis, College of Computing, School of Interactive Computing, Georgia Institute of Technology, Atlanta, GA, USA.
114. Kim, H., & Fellner, D. W. (2004). Interaction with hand gesture for a backprojection wall. In Computer Graphics International, 2004. Proceedings (pp. 395-402). IEEE.
115. Kim, Y., Baylor, A. L., Shen, E. (2007). Pedagogical agents as learning companions: The impact of agent emotion and gender. In Journal of Computer Assisted Learning, 23(3), 220-234.

116. Knobloch, H., Pasamanick, B., Sherard, E. S. (1966). A developmental screening inventory, Department of Pediatrics and Department of Psychiatry, Ohio State University College of Medicine, Columbus, Ohio.
117. Kobsa (2001). Generic user modeling systems. User Modeling and User-Adapted Interaction, 11(1-2), pp. 49-63.
118. Koda, T., Maes, P. (1996). Agents with faces: The effect of personification. In Robot and Human Communication, 1996., 5th IEEE International Workshop on (pp. 189-194).
119. Kojima, M., Sugimoto, M., Nakamura, A., Tomita, M., Inami, M., Nii, H. (2006). Augmented Coliseum: An Augmented Game Environment with Small Vehicles. In Tabletop (pp. 3-8).
120. Korozi, M., Leonidis, A., Margetis, G., Koutlemanis, P., Zabulis, X., Antona, M., & Stephanidis, C. (2012). Ambient educational mini-games. In Proceedings of the International Working Conference on Advanced Visual Interfaces (pp. 802-803). ACM.
121. Lehman, J. F. (1998). Toward the use of speech and natural language technology in intervention for a language-disordered population. Third International ACM Conference on Assistive Technologies.
122. Leong, C.K. (1999). Psychological and educational aspects of Specific Learning Difficulties. Journal of Paediatrics, 4, pp. 151-159.
123. Leuteritz, J. P. , Widlroither, H. , Mourouzis, A., Panou, M., Antona, M. , Leonidis, A. (2009). Development of Open Platform Based Adaptive HCI Concepts for Elderly Users. In C. Stephanidis (Ed.) Proceedings of 13th International Conference on Human-Computer Interaction (HCI International 2009), San Diego, California, USA, 19–24 July. Berlin: Springer.
124. Liarokapis F. (2004). Web3D and augmented reality to support engineering education. World Transactions on Engineering and Technology Education, 3(1), pp. 11-14.
125. Loenen, E.J. (2003) "Ambient intelligence: Philips' vision". Presentation at ITEA 2003, Oulu, Finland, January 10
126. Lund, H. H., Vesisenaho, M. (2004). I-Blocks for ICT Education Development - Case Iringa Tanzania. In Proceedings of 33th International Symposium on IGIP IEEE / ASEE (pp. 364–371). Switzerland: University of Applied Science of Western Switzerland.
127. Macintyre, C. Macintyre, C. (2004). Identifying and supporting children with Specific Learning Difficulties: a holistic approach. USA: Routledge Ed.
128. Maguire M., (2001). Methods to support human-entered design, International Journal of Human-Computer Studies, Volume 55, Issue 4, (October 2001), pp. 587-634, Elsevier Science B.V.
129. Maguire, M., Bevan, N. (2002) User requirements analysis: A review of supporting methods. Proceedings of IFIP 17th World Computer Congress, Montreal, Canada, 25-30 August 2002, p133-148. Kluwer Academic Publishers
130. Manresa, C., Varona, J., Mas, R., Perales, F. (2005). Hand tracking and gesture recognition for human-computer interaction. Electronic letters on computer vision and image analysis, 5(3), 96-104.
131. Marco, J., Cerezo, E., Baldassarri, S. (2013). Bringing tabletop technology to all: evaluating a tangible farm game with kindergarten and special needs children. Personal and ubiquitous computing, 17(8), 1577-1591.
132. Margetis, G., Grammenos, D., Zabulis, X., Stephanidis, C (2013). iEat: An Interactive Table for Restaurant Customers' Experience Enhancement. In C. Stephanidis (Ed.), HCI International 2013 - Posters' Extended Abstracts, Part II - Volume 29 of the combined Proceedings of HCI International 2013 (15th International Conference on Human-Computer Interaction), Las Vegas, Nevada, USA, 21-26 July, pp. 666-670.

133. Margetis, G., Zabulis, X., Koutlemanis, P., Antona, M., Stephanidis, C., (2012). Augmented interaction with physical books in an Ambient Intelligence learning environment. Multimedia Tools and Applications.
134. Margetis, G., Zabulis, X., Ntoa, S., Koutlemanis, P., Papadaki, E., Antona , M., Stephanidis, C (2014). Enhancing education through natural interaction with physical paper. To appear in Universal Access in the Information Society, Special Issue on "User Experience and Access using Augmented and Multimedia Technologies".
135. Maybury, M. T. (1993). Intelligent Multimedia Interfaces. Menlo Park, CA: AAAI Press.
136. Maybury, M., Wahlster, W., (1998). Readings in Intelligent User Interfaces, Morgan Kaufmann Press.
137. McFarlane, A., Sparrowhawk, A., Heald, Y. (2002). Report on the educational use of games: an exploration by TEEM of the contribution which games can make to the education process. Teem, Cambridge
138. Mehrabian, A. (1968). Communication without words. In Psychological today, 2, 53-55.
139. Michalewicz, Z., Michalewicz, M., (2007). Puzzle-based learning. Proceedings of the 18th Conference of the Australasian Association for Engineering Education, Melbourne, Australia
140. Michou, M. (2008), A Semantics-based User Model for the Support of Personalized, Context-Aware Navigational Services, Master’s Thesis, University of Crete Computer Science Department
141. Michou, M., Bikakis, A., Patkos, T., Antoniou, G., Plexousakis, D. (2009). A Semantics-Based User Model for the Support of Personalized, Context-Aware Navigational Services. In First International Workshop on Ontologies in Interactive Systems, 2008. ONTORACT '08. pp 41-50.
142. Mikropoulos, T. A., Strouboulis, V. Factors that influence presence in virtual learning environments. CyberPsychology & Behavior 7, 5 (2004), 582-591.
143. Milgram, P., and Kishino, F. (2004). A taxonomy of mixed reality visual displays, IEICE Transactions on Information and Systems, Special issue on Networked Reality
144. Mills, S., and Araújo, M. M., (1999). Learning through virtual reality: a preliminary investigation. Interacting with Computers 11, 4 pp. 453-462.
145. Mitchell, A., Savill-Smith, C., (2004). The use of computer and video games for learning. A review of the literature. Learning and Skills Development Agency, Ultralab, M-learning
146. Msall, M. E., & Msall, E. R. (2007). Functional assessment in neurodevelopmental disorders. In P. J. Accardo (Ed.), Capute and Accardo's neurodevelopmental disabilities in infancy and childhood (3 ed., pp. 419-443). Baltimore MD: Paul Brookes.
147. Munro BH. (1997). Statistical Methods for Health Care Research. Philadelphia, Pa: Lippincott-Raven
148. Murray-Branch, J. Gamradt, J. E. (1999). Assistive technology: Strategies and tools for enhancing the communication skills of children with Down syndrome. In J.F. Miller, M. Leddy and L.A. Leavitt (Eds.), Improving the Communication of People with Down Syndrome. (pp. 161-204). Baltimore, MD: Paul H. Brookes Publishing Co.
149. Murray-Branch, J. Gamradt, J. E. (1999). Assistive technology: Strategies and tools for enhancing the communication skills of children with Down syndrome. In J.F. Miller, M. Leddy and L.A. Leavitt (Eds.), Improving the Communication of People with Down Syndrome. (pp. 161-204). Baltimore, MD: Paul H. Brookes Publishing Co.

150. Nakanishi Y, Oka K., Kuramochi M., Matsukawa S, Sato Y., Koike H., (2002). Narrative Hand: Applying a fast finger-tracking system for media art
151. Nass, C., Steuer, J., Tauber, E. R. 1994. Computers are social actors. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 72-78). ACM.
152. National Health Service (1998). Information for health: an information strategy for the modern NHS 1198-2005. National Health Service Executive, Department of Health, London
153. Nguyen, T.H.D., Qui, T.C.T., Xu, K., Cheok, A.D., Teo, S.L., Zhou, Z.Y., Mallawaarachchi, A., Lee, S.P., Liu, W., Teo, H.S., Thang, L.N., Li, Y., Kato, H. (2005). Real-time 3D human capture system for mixed-reality art and entertainment. IEEE Transactions on Visualization and Computer Graphics 11, 6 pp. 706-721.
154. Nijholt, A. (2003). Disappearing computers, social actors and embodied agents. In Cyberworlds, (2003). Proceedings. 2003 International Conference on (pp. 128-134). IEEE.
155. O'Hara, M. (2008). Young children, learning and ICT: A case study in the UK maintained sector. Technology, pedagogy and education, 17(1), 29-40.
156. Ohno, T., Mukawa, N., Yoshikawa, A. (2002). FreeGaze: a gaze tracking system for everyday gaze interaction. In Proceedings of the 2002 symposium on Eye tracking research & applications (pp. 125-132). ACM.
157. Oka, K., Sato, Y., Koike, H. (2002). Real-time tracking of multiple fingertips and gesture recognition for augmented desk interface systems. In Automatic Face and Gesture. Recognition, 2002. Proceedings. Fifth IEEE International Conference on (pp. 429-434). IEEE.
158. P. Yellowlees, P. Brooks, (1999). Health online: the future isn't what it used to be, Med. J. Aust. 171 pp. 522–525.
159. Panksepp, Jaak. (2007). Can play diminish ADHD and facilitate the construction of the social brain? Journal of the Canadian Academy of Child and Adolescent Psychiatry, 16, 57–66.
160. Papagiannakis, G. (2013). Geometric algebra rotors for skinned character animation blending. In SIGGRAPH Asia 2013 Technical Briefs, 11.
161. Papagiannakis, G., Papanikolaou, P., Greassidou, E., Trahanias, P. (2014). glGA: an OpenGL Geometric Application framework for a modern, shader-based computer graphics curriculum. In Eurographics 2014, Education Papers, 1–8.
162. Papagiannakis, H., Antona, M., Ntoa, S., Stephanidis, C (2013). A Multimodal Ambient Intelligence Environment for Playful Learning. Journal of Universal Computer Science, special issue on "Towards Sustainable Computing through Ambient Intelligence", 19 (17), 2617-2636.
163. Papagiannakis, H., Ntoa, S., Antona, M., Stephanidis, C. (2012). Learning by playing in an ambient intelligent playfield. In Ubiquitous Computing and Ambient Intelligence (pp. 486-498). Springer Berlin Heidelberg.
164. Partarakis N., (2005). Using Ambient Intelligence Technologies for Producing and Disseminating Art. PhD thesis. Computer Science Department, University of Crete
165. Patrick, E., (2003) Intelligent user Interfaces, Introduction and survey, Research Report DKS03-01 / ICE 01, Version 0.91
166. Perry, E., (2009). The value of play. London: Continuum. p. 85
167. Philips Research (2005): http://www.research.philips.com/technologies/syst_softw/ami/index.html\
168. Philips, Interactive tablecloth, 2006
169. Phillips Research, Ambient intelligence: Changing lives for the better, 2007. www.research.phillips.com/
170. Piaget, J. (1953) How children form mathematical concepts. SciAm 189(5):74-79

171. Piaget, J. (1962). Play, dreams, and imitation in childhood. New York: Norton. Inhelder, B. & Piaget, J. (1964). The early growth of logic in the child. New York: Harper & Row.
172. Piaget, J. (1973). Main Trends in Psychology. London: George Allen & Unwin.
173. Piper, A. M., O'Brien, E., Morris, M. R., Winograd, T. (2006). SIDES: a cooperative tabletop computer game for social skills development. In Proceedings of the 2006 20th Anniversary Conference on Computer Supported Cooperative Work
174. Play Wales http://www.playwales.org.uk/downloaddoc.asp?id=1andpage=67andskin=0
175. Prendinger, H., Ma, C., Yingzi, J., Nakasone, A., Ishizuka, M. (2005). Understanding the effect of life-like interface agents through users' eye movements. In Proceedings of the 7th international conference on Multimodal interfaces (pp. 108-115). ACM.
176. Prensky, M. (2001). Digital game-based learning. McGraw-Hill, New York
177. Ramos, C. (2007). Ambient intelligence - a state of the art from artificial intelligence perspective, In Proceedings of the aritficial intelligence 13th Portuguese conference on Progress in artificial intelligence (EPIA'07), Springer-Verlag, Berlin, Heidelberg, pp. 285-295.
178. Rekimoto, J. (1995). The magnifying glass approach to augmented reality systems. Proceedings of ICAT '95, Makuhari Meese
179. Rekimoto, J., Nagao, K. (1995). The world through the computer: Computer augmented interaction with real world environments. Proceedings of the 1995 User Interface Software Technology, Pittsburgh, 29-36.
180. Research Group Embedded Interaction, Particle Display Website http://www.hcilab.org/projects/particles/particles-display-add-on.htm
181. Revelle G., Druin A., Zuckerman O., Bolas M. (2005). Tangible User Interfaces for Children; Special Interest Groups; Portland USA; CHI 2005
182. Reynolds, P.C. (1976). Play, language and human evolution. In J.S. Bruner, A. Jolly, & K. Sylva (Eds.), Play: Its role in development and evolution (pp. 621–635). New York: Basic Books
183. Riva G, Vatalaro F., Davide F., Alcaniz M. (2005). New Technologies for Ambient Intelligence; IOS Press.
184. Riva S., Antonietti A. (2010). The Application of the ICF CY Model in Specific Learning Difficulties:A Case Study; Psychology of Language and Communication 2010, Vol.14, No. 2; Catholic University of the Sacred Heart in Milan.
185. Rizzo, A. (2002). Virtual reality and disability: Emergence and challenge. Disability and Rehabilitation, 24, pp. 567-569.
186. Roschelle, J.M., Pea, R.D., Hoadley, C.M., Gordin, D.N., Means, B.M. (2000). Changing how and what children learn in school with computer-based technologies. The Future of Children: Children and Computer Technology 10(2), pp. 76–101
187. Roussou, M. (2004) Learning by doing and learning through play: An exploration of interactivity in virtual environments for children. ACM Computers in Entertainment 2, 1 pp. 10- 32.
188. Roy, S. K. Das Bhaumik, A. Bhattacharya, K. Basu, D. J. Cook, S. K. Das, (2003). Location aware resource management in smart homes, Proc. IEEE Int. Conf. Pervasive Computing and Communications (PerCom ’03), pp. 481-488
189. Sailor, M. J., Link, J. R. (2005). “Smart dust”: nanostructured devices in a grain of sand. Chemical Communications, (11), 1375-1383.
190. Salah, A. A., Schouten, B. A., Göbel, S., Arnrich, B. (2014). Playful interactions and serious games. Journal of Ambient Intelligence and Smart Environments, 6(3), pp. 259-262.
191. Salkind, Neil J. (2004). An Introduction to Theories of Human Development SAGE Publications Inc: California.

192. Sarama, J. (2003). Technology in early childhood mathematics: Building Blocks as an innovative technology-based curriculum. Buffalo and Wayne State: State University of New York.
193. Savage, D. (2004). Is everyone keeping up? London: ATL.
194. Savidis, A., Antona, M., & Stephanidis, C. (2005). A Decision-Making Specification Language for Verifiable User-Interface Adaptation Logic. International Journal of Software Engineering and Knowledge Engineering.
195. Scarlatos L. L. (1999). Puzzle Piece Topology: Detecting Arrangements in Smart Objects Interfaces; Brooklyn College
196. Schaper, L. K., Pervan, G. P. (2007). ICT and OTs: A model of information and communication technology acceptance and utilisation by occupational therapists. International journal of medical informatics, 76, S212-S221.
197. Schery, T. K., O'Connor, L. C. (1995). Computers as a context for language intervention. In M.E. Fey and J. Windsor (Eds.), Language intervention: Preschool through the elementary years. (pp.275 -312). Baltimore, MD: Paul H Brookes Publishing Co.
198. Schmidt A., (2005) Interactive Context-Aware Systems Interacting with Ambient Intelligence, IOS Press, http://www.ambientintelligence.org
199. Schmidt, A., Beigl, M., Gellersen, H. W. (1998) There is More to Context than Location, in Interactive Applications of Mobile Computing, Rostock, Germany, 24-25.
200. Schouten, B. A., Tieben, R., van de Ven, A., & Schouten, D. W. (2011). Human behavior analysis in ambient gaming and playful interaction. In Computer Analysis of Human Behavior (pp. 387-403). Springer London.
201. Schumm, J., et al., (2010). "Unobtrusive physiological monitoring in an airplane seat," Personal and Ubiquitous Computing, vol. 14, no. 6, pp. 541–550.
202. Shafer S., Krumm J., Brumitt B., Meyers B., Czewrwinski M., Robbins D. (1998). The New EasyLiving Project at Microsoft Research; Microsoft Corporatopn, Microsoft Research, Gaithersburg; Smart Spaces Workshop 1998.
203. Shaker M., Smith M. N. R., Yue S., Duckett T., (2010). Vision-Based Lading of a Simulated Unmanned Aerial Vehicle with Fast Reinforcement Learning, Lincoln University, UK
204. Shelley E. Mulligan (2003). Occupational Therapy Evaluation for Children, "Table 3-2: Development of Play Skills", pp. 101 - 104.
205. Shevell, M., et al. (2003). "Practice parameter: Evaluation of the child with global developmental delay." Neurology 60
206. Simeonsson R. J., Leonardi M., Bjock-Akesson E. B. Hollenweger J., Lollar D., Martinuzzi A., TenNapel H. (2006). ICF-CY: A Universal Tool for Practice Policy and Research; Tunisia
207. Siraj-Blatchford, J., Siraj-Blatchford, I. (2002). Guidance for practitioners on appropriate technology education in early childhood. Retrieved April 26, 2005, from http://www.ioe.ac.uk/cdl/datec
208. Snowling, M.J. (2005). Specific learning difficulties. Psychiatry, 4, 110-113
209. Srivastava M., Muntz R., Potkonjak M. (2001). Smart Kindergarten: Sensor-based Wireless Networks for Smart Developmental Problem-solving Environments; UCLA, Computer Science Department
210. Stanford, V., (2004) Biosignals offer potential for direct interfaces and health monitoring, IEEE Pervasive Computing 3 pp. 99-103.
211. Stanton, D. et al. (2001). Classroom collaboration in the Design of Tangible Interfaces for Storytelling, In Proc. CHI 2001, ACM Press, pp. 482-489.
212. Stephanidis, C. (2006) A European Ambient Intelligence Research Facility at FORTH - ICS, ERCIM NEWS, pp. 31

213. Stephanidis, C. (2012). Human Factors in Ambient Intelligence Environments. In G. Salvendy(Ed.), Handbook of Human Factors and Ergonomics (4th Edition), Chapter 49 (pp.1354-1373). USA: John Wiley and Sons.
214. Streitz, N., Nixon, P. (2005). The disappearing computer. Communications of the ACM, 48(3), pp. 32-35.
215. Streitz, N., Prante, T., Roecker, C., Van Alphen, D., Stenzel, R. Magerkurth, C., Lahlou, S., Nosulenko, V., Jegou, F., Sonder, F., and Plewe, D. (2007). Smart artefacts as affordances for awareness in distributed teams. (pp. 3-29). Springer Berlin Heidelberg.
216. Stuart, R. (2001). The Design of Virtual Environments. BarricadeBooks Inc., Ft. Lee, New Jersey.
217. Stuberg, W. A., White, P. J., Miedaner, J. A., Dehne, P. R. (1989). Item reliability of the Milani-Comparetti motor development screening test. Physical therapy, 69(5), pp. 328-335.
218. Sukaviriya, P., Foley, J (1993): Supporting Adaptive Interfaces in a knowledge-based user Interface Environment, In W. D. Gray, W. E. Hefley, and D. Murray (Eds.), Proceedings of the 1993 International Workshop on Intelligent User Interfaces (pp. 107-114), Orlando, FL. New York: ACM Press.
219. Sung, M. Pentland, A. MIThril LiveNet (2004). Health and Lifestyle Networking, Workshop on Applications of Mobile Embedded Systems (WAMES'04) at Mobisys'04, Boston, MA, June, 2004
220. Sylva, K., Bruner, J.S., Genova, P. (1976). The role of play in the problem-solving of children 3–5 years old. In J.S. Bruner, A. Jolly, & Sylva, K. (Eds.), Play: Its role in development and evolution (pp.244–261). New York: Basic Books
221. Szekely P., (1991) Structuring Programs to Support Intelligent Interfaces, in Sullivan J.W., Tyler S.W. (eds), Intelligent User Interfaces, ACM Press, pp 445-464.
222. Tanenhaus, J. (1995). Computer learning in early elementary and postsecondary education. In L. Nadel and D. Rosenthal (Eds), Down syndrome: Living and Learning in the Community, (pp.197-201) New York, NY: Wiley Liss.
223. Tartaro, A. (2005). Storytelling with a virtual peer as an intervention for children with autism.
224. Taylor, M. J., McCormick, D., Impson, R., Shawis, T., & Griffin, M. (2011). Activity Promoting Gaming Systems in Exercise and Rehabilitation. Journal of rehabilitation research and development, 48(10), 1171-1186.
225. Terrenghi, L., Kranz, M., Holleis, P. Schmidt, A. (2006). "A cube to learn: a tangible user interface for the design of a learning appliance", presented at Personal and Ubiquitous Computing, pp.153-158.
226. Turbill, J. (2001). A researcher goes to school: Using technology in the kindergarten literacy curriculum. Journal of Early Childhood Literacy, 1(3), 255–278.
227. Ugarte, A., García, I., Ortiz, A., Oyarzun, D. (2007). User interfaces based on 3D avatars for interactive television. In Interactive TV: a Shared Experience (pp. 107-115). Springer Berlin Heidelberg.
228. Valli, A. (2008).The design of natural interaction. Multimedia Tools and Applications. 38(3): 295 - 305.
229. Vanello N., Hartwig V., Tesconi M., et al (2008). Sensing Glove for Brain Studies: design and assessment of its compatibility for fMRI with a robust test; Interdepartmental Res. Center “E. Piaggio”; University of Pisa, Pisa.
230. Vygotsky, L. S. (1978). Mind in society: The development of higher psychological processes. Cambridge, MA: Harvard University Press.
231. Walter, M., Eilebrecht, B., Wartzek, T., Leonhardt, S. (2011). “The smart car seat: personalized monitoring of vital signs in automotive applications,” Personal and Ubiquitous Computing, vol. 15, no. 7, pp. 707–715.

232. Wang, Y. M., Li, Y., Zheng, J. B. (2010). A camera calibration technique based on OpenCV. In Information Sciences and Interaction Sciences (ICIS), 2010 3rd International Conference on (pp. 403-406). IEEE.
233. Weiser, M. (1993) Hot Topics: Ubiquitous Computing, IEEE Computer, October 1993
234. Weiser, M. (1993). Some computer science issues in ubiquitous computing. Communications of the ACM 36 7, pp. 75-84
235. Westeyn T. L., Abowd G. D., Starner T. E., Johnson J. M., Presti P. W., Weaver K. A. (2011). Monitoring children's developmental [rpgress using augmented toys and activity recognition; London
236. Westeyn T. L., Kientz J. A., Starner T. E. Abowd G. D. (2008). Designing Toys with Automatic Play Characterization for Supporting the Assessment of a Child's Development; College of Computing; Georgia Institute of Technology, Atlanta
237. WHO (1980) International Classification of Impairments, Disabilities, and Handicaps. Geneva: World Health Organization
238. WHO (2001) International Classification of Functioning, Disability and Health. Geneva: World Health Organization
239. Winn, W. (1997). The Impact of Three-Dimensional Immersive Virtual Environments on Modern Pedagogy. HITL Technical Report R-97-15. Discussion paper for NSF Workshop. University of Washington, Seattle, WA: Human Interface Technology Laboratory].
240. Wolffenbuttel, R.F., Mahmoud, K.M., Regtien, P.L. (1990). Compliant capacitive wrist sensor for use in industrial robots, IEEE Transactions on Instrumentation and Measurements 39 pp. 991-997.
241. Woods, E., (2004). et al. Augmenting the science centre and museum experience. In proceeding of 2nd international conference on computer graphics and interactive techniques in Australasia and South East Asia pp. 230-236.
242. World Federation of Occupational Therapists, What is Occupational Therapy? WFOT, 2003.
243. World Health Organization (1992). The ICD-10 Classification of Mental and Behavioural Disorders, Geneva: CH, WHO Ed.
244. World Health Organization (2001). ICF the International Classification of Functioning, Disability and Health, Geneva: CH, WHO Ed. Unauthenticated Download Date | 1/5/15 4:21 PM58 Silvia Riva, Alessandro Antonietti
245. World Health Organization (2007). ICF CY the International Classification of Functioning, Disability and Health for Children and Adolescents, Geneva: CH, WHO Ed.
246. Wu, Y. (2009). Research on bank intelligent video image processing and monitoring control system based on OpenCV. In Anti-counterfeiting, Security, and Identification in Communication, 2009. ASID 2009. 3rd International Conference pp. 211-214. IEEE.
247. Y. G. Lim, K. K. Kim, S. Park, (2006). "Ecg measurement on a chair without conductive contact," Biomedical Engineering, IEEE Transactions on, vol. 53, no. 5, pp. 956–959.
248. Yang, S. (2003). "Researchers create wireless sensor chip the size of glitter". UCBerkeley News, June 4, http://www.berkeley.edu/news/media/releases/2003/06/04_sensor.shtml
249. Zhou, F., Been-Lirn Duh, H., Billinghurst, M. (2008). Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR. pp. 193-202.
250. Zimmerman, F. J., Gilkerson, J., Richards, J. A., Christakis, D. A., Xu, D., Gray, S., Yapanel, U. (2009). Teaching by listening: The importance of adult-child conversations to language development. Pediatrics, 124(1), 342-349.

APPENDIX I – ACTA grammar

```
grammar Acta;

// Options
options
{
        language = 'CSharp2';
        output=AST;
        ASTLabelType=CommonTree;
        backtrack=true;
}

// Tokens
tokens
{
        NEGATE;
        CALLARGS;
        FUNCARGLIST;
        MYSTATE;
        GLB_STATELIST;
        WHENSTMTATTR;
        WHENID;
        REEVALUATION;
        PRIORITY;
        WHEN = 'when';
        ASS = '=';
        LT = '<';
        LE = '<=';
        GT = '>';
        GE = '>=';
        EQ = '==';
        NE = '!=';
        AND = '&&';
        OR = '||';
        ADD = '+';
        SUB = '-';
        MUL = '*';
        DIV = '/';
        MOD = '%';
        NOT = '!';
        CALL = '(';
}

// Extra parser membersSTMTLIST
@members
{
        public struct ErrorMsg
        {
                public string Msg;
                public string Pos;
        }

        List<ErrorMsg> m_errors = new List<ErrorMsg>();

        public override void ReportError(RecognitionException e)
        {
                ErrorMsg msg;
                msg.Msg = this.GetErrorMessage(e, this.TokenNames);
                msg.Pos = this.GetErrorHeader(e);
                m_errors.Add(msg);
        }

        public List<ErrorMsg> Errors
        {
                get { return m_errors; }
```

```
        }
}

// Extra imports
@header
{
        using System.Text;
        using System.Collections.Generic;
}

// Rules
program : whenstmt* mystate* EOF -> ^( GLB_STATELIST whenstmt* mystate*);
mystate : 'State' STRING '{' stmt2* whenstmt* '}' -> ^(MYSTATE STRING stmt2*
whenstmt*)
        ;

whenstmt : WHEN '(' logicalExpression ')' '{' stmt2* '}' whenstmtattr? ->
^(WHEN logicalExpression stmt2* whenstmtattr?)
        ;

whenstmtattr: 'with' ':' whenid ',' priority ',' reevaluation (','
whenactive)?-> ^(WHENSTMTATTR whenid priority reevaluation whenactive?);

whenid : ('Name' | 'name')! ASS! STRING;

priority : ('Priority' | 'priority')! ASS! NUM;

reevaluation: ('Reevaluation' | 'reevaluation')! ASS! reevaluationAttr;

reevaluationAttr: 'Always' | 'Never';

whenactive: ('Active' | 'active')! ASS! BOOLEAN;

stmt: ';'!
    | expression ';'!
    ;

stmt2: funccall ';'!
     | assignment_expression ';'!
     ;

expression: assignment_expression
          | logicalExpression
          ;

assignment_expression: lvalue ASS^ expression
                     ;

logicalExpression: booleanAndExpression (OR^ booleanAndExpression )*
                 ;

booleanAndExpression: equalityExpression (AND^ equalityExpression)*
                    ;

equalityExpression: relationalExpression ((EQ|NE)^ relationalExpression)*
                  ;

relationalExpression: additiveExpression((LT|LE|GT|GE)^ additiveExpression)*
                    ;

additiveExpression : multiplicativeExpression((ADD|SUB)^
multiplicativeExpression)*
                   ;

multiplicativeExpression: unaryExpression((MUL|DIV|MOD)^unaryExpression)*
                        ;

unaryExpression: NOT^ primaryExpression
```

```
                | SUB primaryExpression -> ^(NEGATE primaryExpression)
                | primaryExpression
                ;

primaryExpression:
                  | funccall
                  | lvalue
                  | '('! logicalExpression ')'!
                  | STRING
                  | NUM
                  | BOOLEAN
                  | NULL
                  ;

funccall: lvalue (CALL^ (expression (','! expression)*)? ')'!)+
        ;
lvalue: ID
      ;

idlist: ID (','! ID)*
      ;

NUM: ('0'..'9')+ ('.' ('0'..'9')+)?;
BOOLEAN: 'true' | 'false' | 'TRUE' | 'True' | 'False' | 'FALSE';
NULL: 'null';
STRING: '"' ( EscapeSequence | (options {greedy=false;} :
~('\u0000'..'\u001f' | '\\' | '"' ) ) )* '"'
      ;
ID: ('a'..'z' | 'A'..'Z' | '_') ('a'..'z' | 'A'..'Z' | '_' | '0'..'9')*
  ;
// Fragments
fragment EscapeSequence: '\\' ('n'|'r'|'t'|'\"'|'\\'|UnicodeEscape)
                       ;
fragment UnicodeEscape: 'u' HexDigit HexDigit HexDigit HexDigit
                      ;
fragment HexDigit: ('0'..'9'|'a'..'f'|'A'..'F')
                 ;
// Ignore white space
//
WS: (' '|'\r'|'\t'|'\u000C'|'\n') {$channel=HIDDEN;}
  ;

LCOMMENT: '//' ( ~('\n'|'\r') )* ( '\n'|'\r'('\n')? )? { $channel=HIDDEN; }
        ;

NESTED_ML_COMMENT: '/*' (options {greedy=false;} : NESTED_ML_COMMENT | . )*
'*/' {$channel=HIDDEN;}
                 ;
```

Figure 128: ACTA antlr grammar

APPENDIX II - Mimesis game: poses description and voice commands

Pose_0, description: I put my hands near my cheek.
Pose_0, voice command: Put your hands near your cheek.

Pose_1, description: I grasp my knees.
Pose_1, voice command: Grasp your knees.

Pose_2, description: I cross my hands and touch my knees.
Pose_2, voice command: Cross your hands and touch your knees.

Pose_3, description: I point at my nose.
Pose_3, voice command: Point at your nose.

Pose_4, description: I lift my hands up.
Pose_4, voice command: Lift your hands up.

Pose_5, description: I stretch my hands.
Pose_5, voice command: Stretch your hands.

Pose_6, description: I catch with my hands, my waist. And I lift my leg in the air.
Pose_6, voice command: Catch your waist with your hands, and lift your leg in the air.

APPENDIX III – Mimesis game: ACTA script

```
//Poses: P_0, P_1, P_2, P_3, P_4, P_5, P_6
//global conditions
///////////////////

when(UserPresent && !SessionIsActive) { NextState = "1.1"; }

when(!UserPresent && SessionIsActive)
{
    SpeakAsync("leavingQuestion");
    NextState = "UserIsLeaving";
}

when(UserTooFar && SessionIsActive)
{
    SpeakAsync("comeCloser");
    NextState = "UserTooFar";
}

when(UserTooClose && SessionIsActive)
{
    SpeakAsync("goBack");
    NextState = "UserTooClose";
}

//start of the game
///////////////////

State "1.0"
{
    when(UserPresent) { NextState = "1.1"; }
}

State "1.1"
{
    StartSession();
    NextState = "1.2";
}

State "1.2"
{
    SpeakAsync("welcome");
    NextState = "1.3";
}

State "1.3"
{
    when(SpeechRecognized == "yes") { NextState = "2.1"; }
    when(SpeechRecognized == "no") { NextState = "1.4"; }
    when(Time == 15 && Count == 1) { NextState = "1.1"; }
    when(Time > 15 && Count != 1) { NextState = "1.4"; }
}

State "1.4"
{
    SpeakAsync("goodbye");
    EndSession();

    when(Time == 10) {  NextState = "1.0"; }
}

//during game
///////////////
```

```
State "2.1"
{
    SpeakAsync("do_as_fast_as_you_can");
    ShufflePoses();
    when(Time == 5) {   NextState = "Level1"; }
}

State "Level1" //SNV: static non verbal information
{
    DemonstrateStaticPose(CurrentPose);
    when(AnimationRecognized == "" + CurrentPose) { NextState = "Congrats";
}
    when(Time == 10) {  NextState = "Level2"; }
}

State "Level2" //ANV: animate to a posture, non verbal information
{
    DemonstrateDynamicPose(CurrentPose);
    when(AnimationRecognized == "" + CurrentPose) { NextState = "Congrats";
}
    when(Time == 10) {  NextState = "Level3"; }
}

State "Level3" //SV: static with verbal information
{
    DemonstrateStaticPose(CurrentPose);
    SpeakAsync(CurrentPose + "_description");
    when(AnimationRecognized == "" + CurrentPose) { NextState = "Congrats";
}
    when(Time == 10) {  NextState = "Level4"; }
}

State "Level4" //AC: animate to a posture with verbal command
{
    DemonstrateDynamicPose(CurrentPose);
    SpeakAsync(CurrentPose + "_command");
    when(AnimationRecognized == "" + CurrentPose) { NextState = "Congrats";
}
    when(Time == 10 && Count == 3) { ResetCount(); NextState = "4"; }
    when(Time == 10) {  NextState = "Congrats"; }
}

State "Congrats"
{
    SpeakAsync(GetRandomExclamation);
    NextState = "2.1.7";
}

State "2.1.7"
{
    NextPose();
    NextState = "3.1";
}

State "3.1"
{
    when(!PosesDidFinish) { NextState = "Level1"; }
    when(PosesDidFinish)
    {
        ShufflePoses();
        NextState = "Level1";
    }
}

//end of the game
//////////////////
```

```
State "4"
{
    SpeakAsync("leaving");
    EndSession();

    when(Time == 20) {  NextState = "1.0"; }
}

//special cases
//////////////

State "UserIsLeaving"
{
    PauseSession();

    when (UserPresent && Time < 5)
    {
        ResumeSession();
        NextState = PreviousState;
    }

    when (UserPresent && Time >=5 && Time < 10) { NextState = "1.1";    }

    when(Time >=10) { NextState = "1.4"; }
}

State "UserTooFar" { when (!UserTooFar) { NextState = PreviousState; } }

State "UserTooClose" { when (!UserTooClose) { NextState = PreviousState; } }
```

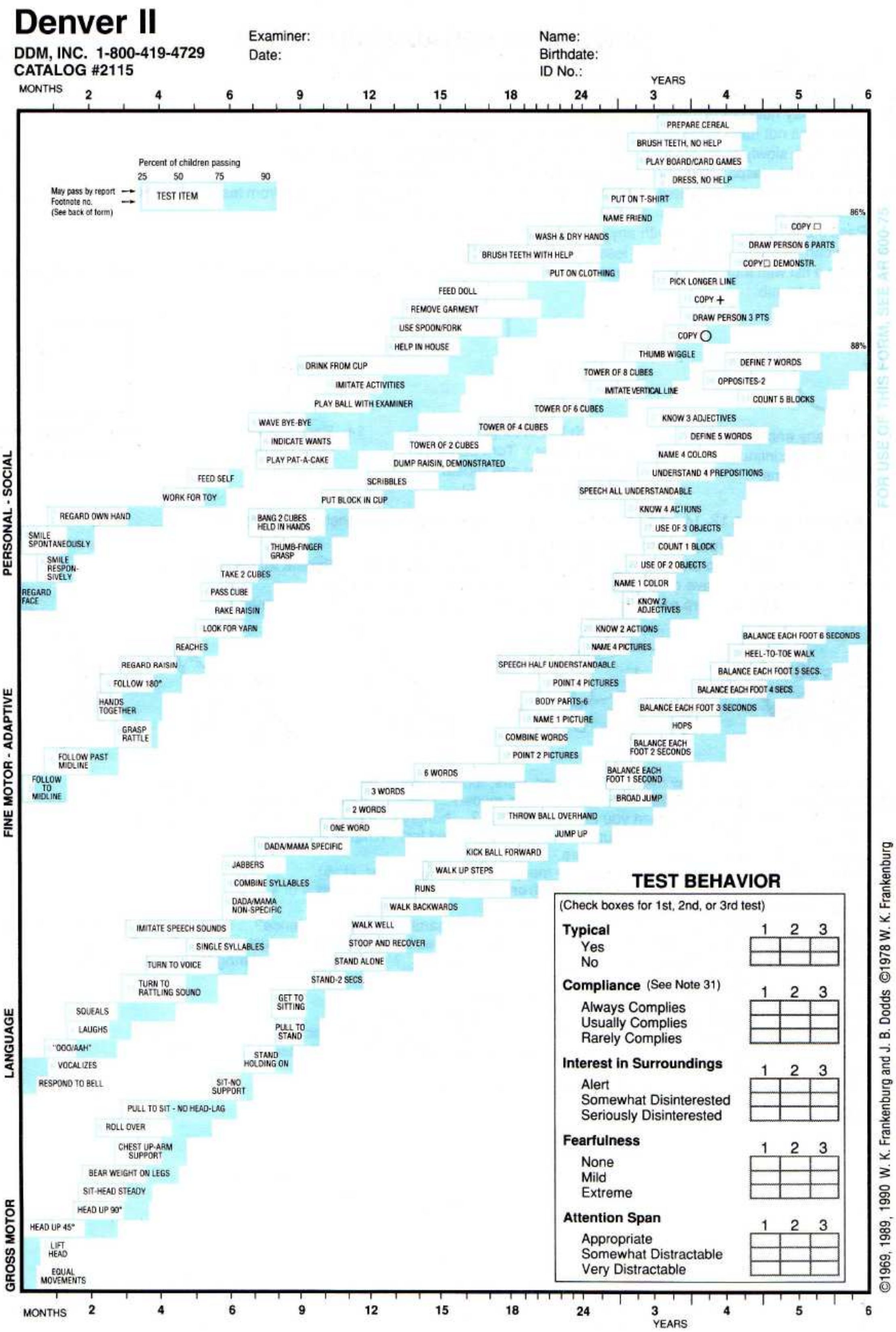
APPENDIX IV - Common Tests and Measures of Development

Developmental Tests and Measures	ICF-CY Classification
Ages & Stages Questionnaire, 3rd edition (ASQ-3)	Activity, Participation
Alberta Infant Motor Scale (AIMS) and Motor Assessment of the Developing Infant	Activity
Assessment, Evaluation, and Programming System for Infants and Children (AEPS), 2nd edition	Activity, Participation
Battelle Developmental Inventory, 2nd edition (BDI-2)	Activity, Participation
Bayley Infant Neurodevelopmental Screener (BINS)	Body structures and functions, Activity
Bayley Scales of Infant and Toddler Development, 3rd edition (BSID-III)	Activity, Participation
Brigance Inventory of Early Development II (BIED-II)	Activity, Participation
Bruininks Oseretsky Test of Motor Proficiency, 2nd edition(BOT-2)	Body structures and functions, Activity
Carolina Curriculum for Infants and Toddlers with Special Needs (CCITSM), 3rd edition, and Carolina Curriculum for Preschoolers with Special Needs (CCPSN), 2nd edition	Activity, Participation
Denver Developmental Screening TestII (DDST-II)	Activity, Participation
Developmental Assessment of Young Children (DAYC)	Activity, Participation
Developmental Hand Dysfunction, 2nd edition	Activity
Developmental Observation Checklist System (DOCS)	Activity, Participation
Developmental Programming for Infants and Young Children—Revised (DPIYC)	Activity
FirstSTEp: Screening Test for Evaluating Preschoolers (FirstSTEp)	Activity, Participation
Harris Infant Neuromotor Test	Activity
Hawaii Early Learning Profile (HELP 0-3 and HELP 3-6), 2nd edition	Activity, Participation
Infanib	Body structures and functions
Infant Motor Profile (IMP)	Activity
Infant-Toddler Developmental Assessment (IDA)—Provence Profile	Activity, Participation
Meade Movement Checklist (MMCL)	Body structures and functions, Activity, Participation
Merrill-Palmer Revised Scales of Development (M-P-R)	Activity, Participation
Milani-Comparetti Motor Development Screening Test, 3rd edition	Body structures and functions, Activity
Miller Assessment of Preschoolers (MAP)	Body structures and functions, Activity
Motor Skills Acquisition in the First Year and Checklist	Activity
Movement Assessment Battery for Children, 2nd edition	Body structures and functions, Activity,
(MOVEMENT ABC-2)	Participation
Movement Assessment of Infants (MAI)	Body structures and functions
Mullen Scales of Early Learning	Activity

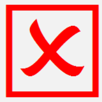
Neonatal Behavioral Assessment Scale, 3rd edition (NBAS-3)	Body structures and functions, Participation
Neonatal Individualized Developmental Care and Assessment Program (NIDCAP)	Body structures and functions, Activity
Neurobehavioral Assessment of the Preterm Infant (NAPI)	Body structures and functions, Activity
Neurological Assessment of the Preterm and Full-term Born Infant, 2nd edition (NAPFI-2)	Body structures and functions
Neurological Exam of the Full Term Infant (NEFTI), 2nd edition	Body structures and functions
Peabody Development Motor Scales, 2nd edition (PDMS-2)	Activity
Posture and Fine Motor Assessment of Infants	Activity
Scales of Independent Behavior—Revised (SIB-R)	Activity, Participation
Test of Gross Motor Development, 2nd edition (TGMD2)	Activity
Test of Infant Motor Performance (TIMP)	Activity
Toddler and Infant Motor Evaluation (TIME)	Activity, Participation
Transdisciplinary Play-Based Assessment, 2nd edition (TPBA2)	Activity, Participation
Vulpe Assessment Battery—Revised (VAB-R)	Body structures and functions, Activity, Participation
Functional Tests and Measures	**ICF-CY Classification**
ABILITIES Index	Body structures and functions, Activity, Participation
Canadian Occupational Performance Measure (COPM), 4th edition	Activity, Participation
Childhood Health Assessment Questionnaire (C-HAQ)	Activity, Participation
Gross Motor Function Measure (GMFM)	Activity
Gross Motor Performance Measure (GMPM)	Body structures and functions, Activity
Pediatric Balance Scale	Body structures and functions, Activity
Pediatric Evaluation of Disability Inventory (PEDI)	Activity, Participation
Pediatric Reach Test	Body structures and functions, Activity
Physical and Neurological Exam for Subtle Signs (PANESS)	Activity
School Function Assessment (SFA)	Activity, Participation
Timed Up and Down Stairs (TUDS)	Body structures and functions, Activity
Timed Up and Go (TUG)	Body structures and functions, Activity
Vineland Adaptive Behavior Scales, 2nd edition (Vineland-II)	Activity, Participation
WEEFIM: Functional Independence Measure for Children	Activity, Participation
Sensory Integration Tests and Measures	**ICF-CY Classification**
Clinical Observations of Motor and Postural Skills (COMPS), 2nd edition	Body structures and functions
DeGangi-Berk Test of Sensory Integration (TSI)	Body structures and functions
Developmental Test of Visual-Motor Integration (VMI), 6th edition	Body structures and functions
Developmental Test of Visual Perception, 2nd edition (DTVP-2)	Body structures and functions
Early Coping Inventory (ECI) and Coping Inventories	Activity, Participation

Infant/Toddler Symptom Checklist	Activity, Participation
Pediatric Clinical Test for Sensory Interaction in Balance (P-CTSIB)	Body structures and function
Sensory Integration and Praxis Tests (SIPT)	Body structures and functions
Sensory Profile and Infant/Toddler Sensory Profile	Body structures and functions
Test of Sensory Function in Infants (TSFI)	Body structures and functions
Test of Visual-Motor Skills-3 (TVMS-3)	Body structures and functions

APPENDIX V - Denver Developmental Screening Test



APPENDIX VI - Questionnaires

Supporting young children in Ambient Intelligence environment. Questionnaire for children for the purposes of the user based evaluation.					
How much did you like the game?	Not at all 	A little 	It was OK 	I liked it 	I liked it a lot 
Have you ever played similar game?	Yes 		No 		
How much different was the game in comparison with the wooden puzzle?	Not at all 	Not that much 	Same 	A bit different 	Very different 
How often would you play the game?	Never again 	Sometime again 	Sometimes 	Often 	Very often 
How hard was it for you to understand how to play?	Very difficult 	Difficult 	It was OK 	Easy 	Very Easy 
How many animals of the farm did you know?	1-2 animals ✱✱	3-4 animals ✱✱ ✱✱	5-6 animals ✱✱✱ ✱✱✱	7-8 animals ✱✱✱✱ ✱✱✱✱	all 9 ✱✱✱✱ ✱✱✱✱✱
Are you satisfied about the duration of the game?	Not at all 	Not that much 	It was OK 	Satisfied 	Very satisfied 

Were there enough things to do while playing?	Not enough	Enough	It was OK	Many	Too many
Did you have enough time between every turn?	Yes			No	
Were the size of the playing board and the game pieces too large, too small or fine for you?	Too small	Small	It was OK	Large	Very large
Did you enjoy the presence of Max?	Not at all	A little	It was OK	I enjoyed it	I enjoyed it a lot
Did you like the colors and the pictures of the game?	Not at all	A little	It was OK	I liked it	I liked it a lot
Do you have any complains?					
How can we make the game better in your point of view?					
Are you curious of what other games you may play with the Beantable;	Yes			No	

Supporting young children in Ambient Intelligence environment. Questionnaire for parents for the purposes of the user based evaluation.

How suitable is for your child the farm game?

0 1 2 3 4 5 6 7 8 9 10

not at all medium very

Did you like the theme of the game?

0 1 2 3 4 5 6 7 8 9 10

not at all medium very

Did you like the use of physical toys while playing?

0 1 2 3 4 5 6 7 8 9 10

not at all medium very

How much did you like the graphics and the presentation of the game?

0 1 2 3 4 5 6 7 8 9 10

not at all medium very

How attractive do you think that the interaction with Max was for your child?

0 1 2 3 4 5 6 7 8 9 10

not at all medium very

Was it easy for your child to play without the presence of an adult?

0 1 2 3 4 5 6 7 8 9 10

not at all medium very

Was the available time for playing satisfactory for your child?

0 1 2 3 4 5 6 7 8 9 10

not at all medium very

Were there enough options for adapting the game to your child abilities?

0 1 2 3 4 5 6 7 8 9 10

not at all medium very

Did the system provide opportunities and support for the development of your child's knowledge and skills?

0 1 2 3 4 5 6 7 8 9 10

not at all medium very

Was the game entertaining for your child?

0 1 2 3 4 5 6 7 8 9 10

not at all medium very

Was there any safety issues during the child's interaction with the system?

0 1 2 3 4 5 6 7 8 9 10

not at all medium very

How different is the farm game compared with other games that you are aware of?

0 1 2 3 4 5 6 7 8 9 10

not at all medium very

Supporting young children in Ambient Intelligence environment. Questionnaire for early intervention professionals for the purposes of the user based evaluation.

Were the goals of the game clearly presented to the child?		
0 1 2 3 4 5 6 7 8 9 10		
not at all	medium	very

Are there enough short-term goals during the game to ensure the entertainment of the child?		
0 1 2 3 4 5 6 7 8 9 10		
not at all	medium	very

How informative was the demonstration of the game?		
0 1 2 3 4 5 6 7 8 9 10		
not at all	medium	very

Was the game entertaining enough to play it again?		
0 1 2 3 4 5 6 7 8 9 10		
not at all	medium	very

Does the game develop the child's skills, which are important for playing the game properly later on?		
0 1 2 3 4 5 6 7 8 9 10		
not at all	medium	very

Were the failures of the child faced in a positive way?		
0 1 2 3 4 5 6 7 8 9 10		
not at all	medium	very

Was the child properly supported for overcoming failures without feeling that he/she is punished?

0 1 2 3 4 5 6 7 8 9 10

not at all medium very

Did the game gradually increase the child's abilities and knowledge?

0 1 2 3 4 5 6 7 8 9 10

not at all medium very

Was the level of the game corresponding to the age range it was designed for?

0 1 2 3 4 5 6 7 8 9 10

not at all medium very

Does the game challenges the child enough in order to continue playing?

0 1 2 3 4 5 6 7 8 9 10

not at all medium very

How interesting was Max for the child?

0 1 2 3 4 5 6 7 8 9 10

not at all medium very

Was the theme of the game proper for the age range it is proposed for?

0 1 2 3 4 5 6 7 8 9 10

not at all medium very

Were system's reactions consistent, challenging and exciting enough for the player?
0 1 2 3 4 5 6 7 8 9 10
not at all medium very

How immediate was the feedback received corresponding to the child's movements?
0 1 2 3 4 5 6 7 8 9 10
not at all medium very

Was enough information provided in order to get the game started?
0 1 2 3 4 5 6 7 8 9 10
not at all medium very

How intrusive was Max during the play?
0 1 2 3 4 5 6 7 8 9 10
not at all medium very

How informative was the help provided during the game for the player in order to move on in case of facing difficulties?
0 1 2 3 4 5 6 7 8 9 10
not at all medium very

Was the application of Bean Administration Suite easy to use?
0 1 2 3 4 5 6 7 8 9 10
not at all medium very

Do you believe that this application is useful / necessary?		
	0 1 2 3 4 5 6 7 8 9 10	
not at all	medium	very

Do you like Bean Administration Suite?		
	0 1 2 3 4 5 6 7 8 9 10	
not at all	medium	very

Do you have any specific recommendations that you believe they would improve the game?

Thank you!

APPENDIX VII – Article in the newspaper “Kathimerini”

16 • Η ΚΑΘΗΜΕΡΙΝΗ ΑΠΟ ΑΛΛΗ ΣΚΟΠΙΑ Παρασκευή 13 Δεκεμβρίου 2013

ΚΑΘΗΜΕΡΙΝΑ

Του Νικου Γ. Ξυδακη

Τραπεζική πίστη σε καιρό κρίσης

Οι Ελληνες καταθέτες έχουν υποφέρει πολύ τα τελευταία χρόνια. Φήμες, φόβοι, ανασφάλεια, υποχρεώσεις οικογενειακές, έκτακτοι φόροι που γίνονται μόνιμοι, ένα παιδί που είναι άνεργο, όλα αυτά έχουν ροκανίσει τις καταθέσεις. Το χειρότερο όλων, όμως, είναι ότι έχει φθαρεί και η τραπεζική πίστη και όχι μόνο εξαιτίας φημών.

Το «κούρεμα» καταθέσεων στην Κύπρο, το οποίο έβλαψε και πολλούς Ελλαδίτες, έφερε τη μεγαλύτερη σύγχυση. Το κυπριακό bail in δεν ήταν φήμη ή αδικαιολόγητος πανικός· ήταν επίσημη πράξη του Eurogroup, του πιο αρμόδιου θεσμικού οργάνου της Ευρωζώνης. Και μπορεί οι τράπεζες πράγματι να είχαν παραβεί τα κόκκινα όρια, ωστόσο γι' αυτό δεν ευθύνονταν οι καταθέτες που έχασαν τα χρήματά τους.

Το μήνυμα επαναλαμβάνεται τώρα εμμέσως από τα ευρωπαϊκά όργανα που προχωρούν στην τραπεζική ενοποίηση. Για τη διάσωση των χρεοκοπημένων τραπεζών στην Ευρωζώνη, από το 2016, θα «κουρεύονται» καταρχάς οι μέτοχοι, ομολογιούχοι και καταθέτες, κατά το κυπριακό μοντέλο. Εφόσον δεν επαρκούν τα ίδια μέσα, θα επιστρατεύονται εθνικοί πόροι και, αν δεν επαρκούν κι αυτοί, κοινοί ευρωπαϊκοί πόροι. Υπενθυμίζεται ότι έως σήμερα, στην Ευρωζώνη έχουν εφαρμοστεί όλες οι εκδοχές διάσωσης τραπεζών: στην Ελλάδα επιβαρύνθηκε με 50 δισ. το δημόσιο χρέος για να ανακεφαλαιοποιηθούν οι τράπεζες, στην Ισπανία διατέθηκαν περίπου 41 δισ. από τον ESM απευθείας στις τράπεζες χωρίς επιβάρυνση του δημόσιου χρέους (σύμφωνα με την ιστορική απόφαση της συνόδου κορυφής τον Ιούνιο 2012), τέλος, στην Κύπρο έκανε πρεμιέρα το bail in, δηλαδή το «κούρεμα» των καταθετών ως πρώτο βήμα.

Η απουσία ξεκάθαρης στρατηγικής έχει προκαλέσει σύγχυση στους καταθέτες.

Η απουσία ξεκάθαρης στρατηγικής έχει προκαλέσει σύγχυση στους καταθέτες. Τα χρόνια της κρίσης, πολλά χρήματα πέταξαν από τις ελληνικές τράπεζες προς τράπεζες του εξωτερικού ή προς τα σεντούκια. Κι αυτό δεν είναι εγχώριο φαινόμενο: ακόμη και πριν από το κυπριακό bail in, από τις χώρες της χειμαζόμενης ευρωπεριφέρειας (Ισπανία, Πορτογαλία, Ιρλανδία, Ελλάδα), από τον Ιούλιο 2011 έως τον Ιούλιο 2012, έφυγαν 326 δισ. ευρώ, σύμφωνα με στοιχεία του Bloomberg. Το ίδιο διάστημα, οι καταθέσεις στις τράπεζες του ευρωπαϊκού Βορρά αυξήθηκαν κατά 300 δισ. ευρώ.

Πώς διαμορφώνεται η κατάσταση; Ο καταθέτης που έχει χρήματα σε ελληνική τράπεζα σήμερα εισπράττει πολύ χαμηλό επιτόκιο. Είναι ασφαλής, όμως, έναντι κλοπής, ληστείας κ.λπ. Αν χρωστάει στην εφορία, οποιαδήποτε κατάθεσή του, ακόμη και σε κοινούς λογαριασμούς με παιδιά, γονείς, σύζυγο, είναι ευσπρόσβλητη. Αν κάτι πάει στραβά στις τράπεζες, μπορεί να «κουρευτούν» οι καταθέσεις, ακόμη και κάτω από εγγυημένο όριο των 100.000; Σε αυτό το κρίσιμο ερώτημα, ο διοικητής της Τράπεζας της Ελλάδος απάντησε σιβυλλικά: Στο προβλεπτό μέλλον δεν θα υπάρξει «κούρεμα». Ο Ευρωπαίος επίτροπος Χ. Αλμούνια στο ίδιο ερώτημα απάντησε ότι το «κούρεμα» επί των εγγυημένων καταθέσεων θα το αποφασίσουν οι εθνικές κυβερνήσεις.

Για το μέγα πλήθος Ελλήνων μικροκαταθετών, όμως, η αγωνία του «κουρέματος» μάλλον δεν υπάρχει: Πόσοι έχουν καταθέσεις άνω των 100.000 ευρώ;

Το «τραπέζι-φασόλι», που επιτρέπει την παρακολούθηση της αναπτυξιακής εξέλιξης των παιδιών και το «Πολύτροπο Ταξίδι», ψηφιακός χάρτης που χρησιμοποιεί το Μουσείο Θεσσαλονίκης.

«Διάχυτη νοημοσύνη», το άλλο πρόσωπο της Ελλάδας

Του **ΓΙΑΝΝΗ ΠΑΛΑΙΟΛΟΓΟΥ**

Ηταν 2 το απόγευμα σε μια ασυνήθιστα κρύα και βροχερή μέρα για την Κρήτη. Ο Γιοχάνες Χαν, επίτροπος της Ε.Ε. για θέματα Περιφερειακής Ανάπτυξης και η συνοδεία του, έπειτα από ένα γεμάτο πρωινό επαφών, είχαν φτάσει στο Ιδρυμα Τεχνολογίας και Ερευνας (ΙΤΕ), 10 χιλιόμετρα από το Ηράκλειο. Πρόκειται για το μεγαλύτερο ερευνητικό κέντρο της χώρας, το οποίο η Ευρωπαϊκή Επιτροπή τοποθέτησε στη 15η θέση μεταξύ των καλύτερων ερευνητικών ιδρυμάτων της Ε.Ε. για την περίοδο 2007-12. Πιο συγκεκριμένα, ο κ. Χαν ξεναγούνταν στη μονάδα ambient intelligence («διάχυτης νοημοσύνης») του Ιδρύματος, την οποία συγχρηματοδοτεί η Ε.Ε. (ο προϋπολογισμός του έργου κατασκευής και εξοπλισμού του κτιρίου είναι 4,27 εκατ. ευρώ). Οπως εξήγησε στην «Κ» ο Δημήτρης Γραμμένος, κύριος ερευνητής στο Εργαστήριο Επικοινωνίας Ανθρώπου-Υπολογιστή του ΙΤΕ, που ηγήθηκε της ξενάγησης, η ιδέα της «διάχυτης νοημοσύνης» είναι ότι ο υπολογιστής σταματά να είναι ένα φυσικό αντικείμενο και μετατρέπεται σε διάχυτη παρουσία στον χώρο, που αλληλεπιδρά ποικιλοτρόπως με τον χρήστη.

Ο Αυστριακός επίτροπος δοκίμασε από πρώτο χέρι μια σειρά από τουριστικές αλλά και εκπαιδευτικές εφαρμογές που έχουν αναπτυχθεί σε αυτή τη λογική. Γνώρισε επίσης το ρομπότ του εργαστηρίου ρομποτικής, το οποίο τον καλωσόρισε και χόρεψε έναν κρητικό χορό για λογαριασμό του.

Το ρομπότ του εργαστηρίου υπολογιστικής όρασης και ρομποτικής του ΙΤΕ επιδεικνύει το εύρος του χορευτικού του ταλέντου.

Μεταξύ των εμπνεύσεων που είχε την ευκαιρία να γνωρίσει ο επίτροπος ήταν το bean table, «τραπέζι-φασόλι» σε ελεύθερη μετάφραση, λόγω του ιδιαίτερου σχήματός του. Το τραπέζι απευθύνεται σε παιδιά προσχολικής ηλικίας, 3-7 ετών. Διαθέτει οθόνη αφής στην οποία μπορεί το παιδί να παίξει παιχνίδια μνήμης ή να αλληλεπιδράσει με έναν εικονικό χαρακτήρα που του δίνει διάφορες οδηγίες. Πάνω στην οθόνη, επίσης, εφαρμόζονται διάφορα υλικά παιχνίδια (π.χ. παζλ), με τη «διάχυτη νοημοσύνη» του τραπεζιού να κατευθύνει τη συμπεριφορά του «χρήστη» ανάλογα με τις επιδόσεις του.

Εκεί έγκειται και η καινοτομία του bean table: πέρα από ένα ελκυστικό παιχνίδι, είναι ένα ανεκτίμητο εργαλείο για γονείς αλλά και ειδικούς που ασχολούνται με την ανάπτυξη των παιδιών. Οπως λέει στην «Κ» ο Μάνος Ζηδιανάκης, ο 29χρονος ερευνητής από το διδακτορικό του οποίου προέκυψε το τραπέζι-υπολογιστής, οι αισθητήρες του –συμπεριλαμβανομένων τεσσάρων στο καρεκλάκι και πέντε στην οθόνη– καταγράφουν λεπτομερώς κάθε πτυχή της συμπεριφοράς του παιδιού. Με τον τρόπο αυτό, «μπορεί να εντοπίσει τυχόν μαθησιακές ή αναπτυξιακές δυσκολίες», εξηγεί ο Ζηδιανάκης, για τις οποίες μπορεί να ενημερώσει τους γονείς. Επιπλέον, με το εύρος των στοιχείων που καταγράφει –από τη μέτρηση της πίεσης που ασκείται στην οθόνη, την αναγνώριση φωνής, χειρονομιών και στάσης του σώματος ώς την παρακολούθηση βλέμματος και τον έλεγχο κατανομής του βάρους του παιδιού στο καρεκλάκι– μπορεί να δώσει πολύτιμες πληροφορίες και σε εργοθεραπευτές και παιδοψυχολόγους.

Εντυπωσιασμένος ο επίτροπος για Θέματα Περιφερειακής Ανάπτυξης, Γιοχάνες Χαν, από τις εφαρμογές που αναπτύσσουν επιστήμονες στο Ιδρυμα Τεχνολογίας και Ερευνας στην Κρήτη.

Ενθουσιασμένος απ' όσα είδε, ο επίτροπος Χαν δήλωσε στην «Κ» ότι θα προτείνει να τοποθετηθούν οι εφαρμογές της «διάχυτης νοημοσύνης» στο κτίριο Berlaymont, το αρχηγείο της Κομισιόν στις Βρυξέλλες, κατά τη διάρκεια της ελληνικής προεδρίας. «Θα είναι μια ευκαιρία να αναδειχθεί το άλλο πρόσωπο της Ελλάδας» σημείωσε χαρακτηριστικά ο κ. Χαν, ο οποίος επανειλημμένως στην επίσκεψή του στην Κρήτη τόνιζε τη σημασία διασύνδεσης της έρευνας με την αγορά.

Αν ο εφευρέτης του bean table –που αποτελεί ακόμα πειραματικό πρωτότυπο, αρκετά στάδια μακριά από την εμπορική αξιοποίηση– θελήσει να το μετατρέψει σε επιχειρηματική δραστηριότητα, το ΙΤΕ έχει τα μέσα και τη διάθεση να τον στηρίξει. Οπως μας ενημέρωσε ο Κώστας Φωτάκης, πρόεδρος του ΙΤΕ, στο Πάρκο Επιστήμης και Τεχνολογίας του Ιδρύματος «εκκολάπτονται» σήμερα 20 εταιρείες ερευνητών και φοιτητών, ενώ 45 εταιρείες έχουν γεννηθεί στο Πάρκο την τελευταία δεκαπενταετία. Οπως σημειώνει στην «Κ», η υστέρηση της Ελλάδας στη μετάβαση από την έρευνα στο επιχειρείν οφείλεται σε μια σειρά παραγόντων, από το υψηλό κόστος δανεισμού και κατοχύρωσης εφευρέσεων έως «την απίθανη γραφειοκρατία και το παράλογο πολλές φορές κανονιστικό πλαίσιο».

Στην Κρήτη, πάντως, το υπόδειγμα υπάρχει. Θα το ακολουθήσουμε;