



EXPERIMENTING WITH AN UBIQUITOUS COMPUTING OPEN ARCHITECTURE

Distributed, Embedded and Modular Computing
for Novel Practices of Play and Therapy in the Water

Alessandro Pollini

PH.D. PROGRAM IN
TELEMATICA E SOCIETÀ DELL'INFORMAZIONE
CICLE XX

Instituted by the Italian University Consortium among
University of Florence
University of Siena

EXPERIMENTING WITH AN UBIQUITOUS COMPUTING OPEN ARCHITECTURE

A thesis submitted for the degree of
Doctorate of Philosophy

Candidate:
Alessandro Pollini

Coordinator: Prof. Dino Giuli

Supervisors:
Prof. Patrizia Marti
Prof. Pierluigi Emiliani

The thesis has been supported by PalCom, an integrated project in EU 6th Framework Program under the proactive initiative 'The Disappearing Computer in Future and Emerging Technologies' (FET), part of the Information Society Technologies.

For further reference see: <http://www.ist-palcom.org/>

The duration of the project is 48 months (01.01.2004 - 31.12.2007).

Abstract

This thesis describes the relation between software architectural development and use practices in the field of Ubiquitous Computing. The main objective is the evaluation of an open software architecture that addresses palpable use of technology, i.e. assemblability, adaptability, resource awareness and experimentability. It focuses on middleware architecture for managing, using and controlling distributed, networked and embedded computational resources.

The rationale of this work stems from the fact that current Ubiquitous Computing systems are designed for invisibility, automation and scalability. Despite this assumption was part of the initial fascinating vision of Ubiquitous Computing, it has rarely been accompanied by the concrete possibility for the end-users to understand, control, adapt and customize technology.

The assumption behind this research is that the multitudes of interconnected devices may rely on a common enabling software architecture, and this has to be designed according to end-users' needs.

This work attempts to understand and evaluate the architectural mechanisms required by usable ubiquitous technology. We study if and how *palpable computing* might support the peculiar challenges posed by the psychomotor therapy in the water for children with special needs.

This is done through the Active Surfaces case study. Active Surfaces is a distributed, embedded and modular system developed to support the therapists for the intervention with disabled children.

The research is described in five steps. First the notion of palpability of UbiComp technology is introduced and discussed through the description of architectural qualities. The second step is the understanding of the implications the case study may have, and the current practices of the therapists are analysed. The third and fourth steps are the design and prototyping of Active Surfaces. The development of the open architecture; the different architectural and application prototypes are described. The fifth step is represented by the experimentations. The experimental plan, the methods and the results are described for both the software performance testing and the user testing.

The approach followed integrates ethnographic methods, interaction design and software architecture engineering as different strands that are interwoven in the iterative research process.

Acknowledgements

I would like to express my gratitude to all of those individuals that somehow contributed to this effort.

My gratitude goes first to Prof. Patrizia Marti, my supervisor. Beginning with my Master's thesis, she has supported my work for the last four years with dedication and trust. She has offered guidance and provided motivation with challenging research questions, always pushing me to give my best. I especially thank her for the interest, the enthusiasm and the patience she has always accorded to this work.

I also thank Prof. Pier Luigi Emiliani, Director of CNR-IFAC, Florence for his insightful comments and the interest he showed in this work.

This work wouldn't have been possible without all they who contributed to this research within the PalCom project: Antonio Rizzo at the University of Siena; Mie Shou Olsen and Gunnar Kramp at the Aarhus School of Architecture; Peter Andersen, Jeppe Brønsted, Jakob Frølund, Henrik Gammelmark and Peter Orbaek at the Computer Science Dpt. University of Aarhus; Boris Magnusson and David Svensson Fors at the Computer Science Dpt., University of Lund. Thank you all.

I would also like to thank the therapists at the Functional Rehabilitation Unit, Le Scotte Hospital, Siena, and the director Maria Grazia Burroni; the trainers, therapists and friends of the Parental Disabled Children Association, Siena, especially Laura Cardosi; and the therapists Alessio Persico and Alessandra Picasso at the D. Chiossone Institute, Genova. I extend my gratitude to each of them for their kind collaboration and enthusiasm during the different design activities.

Thanks to Anna Di Biase for the essential contribution to the linguistic elaboration of this manuscript.

I would like to especially thank my colleagues, with whom I have always shared much more than work. My gratitude goes to Alessia Rullo, Erik Gronvall and Leonardo Giusti for their kind-hearted presence, and for all the fruitful and passionate discussions. I would have encountered many difficulties without their support. Alessia and Erik directly contributed to this work, as they were a part of the research group for the duration of the PalCom project. Erik contributed to this research in a particular way by providing his technical aid throughout the preparation for the experiments. Thank you very much.

Then I also want to thank mum, dad and my sister for their unconditional love and support. They have always left me free to find my own way while still remaining constantly by my side.

Last, and most important, I thank the person with whom I am in love, Elisa, the only person who knows how to stand by me and at the same time love me. She has taken care of me throughout this work, always reminding me of the most important things in life.

Table Of Contents

ABSTRACT

ACKNOWLEDGEMENTS

INTRODUCTION

- Research Theme ____3
- Research Objectives ____5
- Organization of the Thesis ____6

1_ ARCHITECTURAL QUALITIES AND USE PRACTICES

- 1.1 Architectural Qualities for palpability ____8
- 1.2 User diversity and Use practices ____13
- 1.3 Research methodology ____16
 - 1.3.1 Methods ____17
 - 1.3.2 Process ____20

2_ PRACTICES OF PLAY AND THERAPY IN THE WATER

- 2.1 Play and Therapy ____24
 - 2.1.1 Conceptual basis ____25
 - 2.1.2 Main objectives ____28
 - 2.1.3 Target users ____29
 - 2.1.4 Two approaches ____32
- 2.2 Enabling technologies ____37
- 2.3 Fieldwork and Activity analysis ____42
 - 2.3.1 Actors, Context and Tools ____42
 - 2.3.2 Activity modeling ____46
- 2.4 Activity scenarios ____53

3_ ACTIVE SURFACES: DESIGN AND DEVELOPMENT

- 3.1 The Therapeutic Play ____60
- 3.2 The concept ____61
- 3.3 Concept evaluation and refinement ____63
- 3.4 Concept development ____66
- 3.5 Envisioning Scenarios ____72

4_ PROTOTYPING ACTIVE SURFACES

- 4.1 Architectural Qualities in Active Surfaces ____79
 - 4.1.1 Aspects of Scalability ____81
- 4.2 The PalCom open architecture ____83
 - 4.2.1 Main concepts ____86
 - 4.2.2 Middleware Managers ____88
 - 4.2.3 Runtime Environment ____89
 - 4.2.4 Execution Platform ____91
- 4.3 Prototyping Active surfaces ____91
 - 4.3.1 Lightweight embedded prototypes ____93
 - 4.3.2 Architectural prototypes ____101
- 4.4 Prototype scenarios ____107

5_ EXPERIMENTATION

- 5.1 Introduction ____114
- 5.2 Objectives ____117
- 5.3 Objective 1 - Performance Testing ____117
 - 5.3.1 Methods ____117
 - 5.3.2 Tasks ____120
 - 5.3.3 Results ____125
 - 5.3.4 Discussion ____135
- 5.4 Objective 2 - User Testing ____138
 - 5.4.1 Methods ____138
 - 5.4.2 Exploratory Trials ____140
 - 5.4.3 Results ____143
- 5.5 Summative Discussion ____147
 - 5.5.1 Future development of the Open Architecture ____147
 - 5.5.2 Architectural Qualities Explored and Revised ____152

6_ CONCLUSIVE DISCUSSION

- 6.1 Ubiquitous Computing in use ____156
- 6.2. Active Surfaces as toy problem ____157
- 6.3. Methods and prototypes ____158
- 6.4 Conclusions and Open Questions ____160

REFERENCES

Introduction

As our demands for services, functions, and features grow, so, too, does the complexity of our devices increase beyond reasonable comprehension (Norman 2007). Cameras, mobile phones, audio players and television equipment, car systems, even some public toilets now require electricity, display panels, and instruction manuals. This complexity often leads to frustration, and challenges users' abilities, especially because it delivers the advanced capabilities of devices, applications and services to individuals. Simultaneously, as described in Norman (2007), the complexity of the underlying systems and architectures has increased dramatically.

The Ubiquitous Computing technology in practice is often characterized by users that experience recurring breakdowns, standards' incompatibility and a proliferation of accessories when using, accessing and trying to connect different devices (e.g. PCs, cameras, printers, and phones). Such interconnected devices populate ordinary Ubiquitous Computing scenarios.

The focus of the present research is on how a Ubiquitous Computing open architecture can be experimented with in small embedded devices and how people use these technology to enhance their practices and reach personal goals. The architecture described in this research is *open* in the sense that it allows for the open-ended composition and development of applications and that it allows for interoperation with services from diverse platforms. Ubiquitous Computing technology is thought to support purposeful and meaningful uses, and this requires a special attention to users' individual needs and perspectives. This focus has already been experimented in the tradition of user-centered design, and will now be inherently experimented and applied to the design of ubiquitous and ambient computing systems, in which technologies are pervasive to human activities. The users' attitudes and motivations are essential to the comprehension of how complex and multifaceted ambient computing systems have been adopted and made sense of.

With the terms *invisible* (Norman 1998), *disappearing* (Wejchert 2000) and *calm computing* (Weiser, Seely Brown 1996), the core of Ubiquitous Computing has been described. These qualities indicate that computing is passing to the background of human activity. Making the computer invisible is about changing how humans perceive the computer; in order to make the computer disappear, the user, while adopting the technology to carry out a task, keeps their focus on the task itself rather than on the tool (Weiser, Seely Brown 1996). In the early vision of Ubiquitous Computing (Weiser 1991; 1993) this process is realized through the design for calm technology: technology that moves from the centre of our attention outward to the periphery, and between those two foci as it is required by the situation at hand (background-foreground). In this way, technology is taught to not overload people with information that requires conscious attention, and is designed to have a calming effect.

Since the Ubiquitous systems are interwoven with the user's actions, the focus of the interaction designer moves toward the description of the relationships between the use, the infrastructures and the environment.

These attributes are the motivation behind the new way of thinking about computing (and computational resources) and their use. Technology that does not require active attention and is ready to be used at a glance becomes part of the natural human environment and "vanishes into the background" (Weiser).

Ubiquitous computing technology is unobtrusively merged with real world objects and places, so that in a sense it disappears into the background. This implies that new functionalities and new ways of using artifacts will enhance users' practices: "Artifacts will be able to adapt and change, not just in a random fashion but based on how people use and interact with them. [...], resulting in an everyday world that is more *alive* and *deeply interconnected*" (Wejchert 2000).

As Norman (2007) has recently observed, the invisible, ubiquitous computer affords many human activities with its presence, which poses wonderful opportunities and challenges to the field of interaction design and

human-computer interaction. He focused on the interaction between users and ubiquitous computing systems and outlined some important concerns to account for the design for users experience. The UbiComp vision experienced in practice has brought the ever-increasing complexity of technology, the ever-increasing burden of security, authentication, and identification and the ever-increasing use of automation as consequences (Norman 2007). Together with Norman, other authors have revised the UbiComp vision put into practice throughout the past decade (Rogers 2006, Suchman 2007). In particular, Rogers (2006) outlines an alternative account for Ubiquitous computing that focuses on the design for *engaging user experiences* and suggests a shift from proactive computing to proactive people, whereas UbiComp technologies are designed not to do things for people but to engage them more actively in what they currently do. In her perspective, ubiquitous and ambient computing should, in fact, promote engaged living, and technology is designed to enable people to do what they want, need or even have never expected (Rogers 2006). These two perspectives deal with the opposite roles played by people and technologies. In one case the automation of UbiComp is predominant (Weiser 1991), in the other, the user experience is given an increasing importance to the detriment of the technology (Rogers 2006).

Designing ubiquitous technology for the material environment also poses challenges regarding the bodily experience. Traditional computational resources in ICT became immediately observable and available through the use of GUI's windows, menus, buttons, icons etc. With the move of computation off the desktop and into the material environment around us, computation is beginning to manifest itself in new ways, i.e. as an aspect of the everyday environment (Dourish, Bell 2007). The phenomenological understanding of human experience (Merleau-Ponty 1962) has been adopted as a framework for investigating the user experience in CSCW (Robertson 2002). In this account human experience is essentially an active perception because of its embodied, spatially and temporally located presence in the living world (Merleau-Ponty, 1962). Thus our bodies play a central role in shaping our interactions with the world, particularly with regard to the abilities and skills we have and how they affect our social exchanges and our interactions with objects and technology. For example, humans possess a wide range of natural motor-sensory abilities and only a few of these skills and abilities have been adequately modelled in the domain of human-computer interaction. Far from the desktop metaphor, interacting within the physical environment applies different rules with relation both to the embodiment of the technology and user physical actions. Physical activity is then interwoven with cognitive structuring (Piaget 1962; Varela et al. 1991; Clark 1997). Moreover, bodily engagement with physical objects also facilitates active learning, motivation and cognitive triggering (Montessori 1912), and, in this research, these are the presuppositions for the conceptual and experimental investigation of Ubiquitous Computing with users with special needs.

Research Theme

In this thesis a new perspective on Ubiquitous Computing is presented, applied and experimented. This research is part of the EU funded integrated project *PalCom, Palpable Computing*, which aims at defining a new perspective on Ambient and Ubiquitous Computing. Palpable computing complements key features of ubiquitous computing, such as invisibility and end-user composition of devices, with their opposites - e.g., visibility and decomposition – to enable users to independently navigate and influence the computing system (Shultz, Corry 2005). The main motivation of the project is, in fact, that palpable computing complements the “unobtrusive effectiveness of ubiquitous computing” (PalCom Project Website).

The main goal of the PalCom project is to provide an enabling *architecture* to make Ubiquitous Computing *palpable* for the users. The PalCom software architecture aims at structuring the computational resources within the environment: their software components, their externally visible properties, and the relationships

between components (PalCom Project Website). Together with software architectures for palpable computing systems, general system components (i.e. devices and hardware) are key components to be handled and accounted for in the architecture.

The PalCom framework has been represented by means of theoretical challenges, which outline the vision of the project with the possibility of choosing the appropriate and advantageous solution for each application case throughout the continuum between the opposites of the dichotomies. It represents, in fact, not a strict choice between the opposites, but rather an opportunity to change and move throughout intermediate positions within the dichotomies.



Fig. 1. Challenges defined in the initial conceptual framework of the PalCom project.

The Palpable Computing is described on the basis of the balance and the adaptation between the two opposites. Notwithstanding the main goal of Ubiquitous Computing applications is to make the technology transparent or invisible (Weiser 1993) to the users, when this is in practice, it becomes obvious that this sort of technological disappearance is not always possible or even desirable. When, for example, an error occurs within ubiquitous systems, a user would benefit from the visibility of the current system state, which would thus permit the inspection of the error and allow, if possible, the necessary correctional measures to be taken.

We can assume as an introductory instance that the natural material environment we live in is immediately *palpable* by definition; it is in fact noticeable, available and plain-to-see. It is immanent and plainly observable and comprehensible, such that we can always experience the potential for action we have within it.

The framework introduced above lets us explain *what palpable computing is* and *what making ubiquitous computing palpable means* in practice.

Research Objectives

- (1) At a conceptual level, this thesis aims to investigate *the use of ubiquitous computational resources, how they are adopted and what is their impact on everyday practices.*
- (2) In particular, this is pursued by *exploring and experimenting with the PalCom open architecture through resource constrained, embedded devices adopted in the Active Surfaces application prototype, which is used in the context of the therapeutic intervention in water on children with special needs.*
- (3) From a methodological perspective the research intends to investigate the way in which *the use of application and architectural prototypes informs the development of the software architecture.*

Active Surfaces is a modular distributed system of embedded devices that have been used as test beds for implementing and testing part of the PalCom software architecture. The architecture has been scaled down and experimented with in small devices by means of resource constrained hardware implementation. In this research a number of ongoing prototypes have been developed and tested with the end users. In fact PalCom embraces everyday work activities and requires that both simulated and embedded prototypes have to be used and experimented with the users. PalCom developed prototypes of future devices and services in collaboration with a number of users in different domains: healthcare, landscape architecture, and emergency response. These prototypes ran on and informed the design of a software architecture that seeks to support for *making computing palpable.*



Figure 2. The Active Surfaces application prototype

The main inspiration of this research thesis derives from the exploration of a specific field within the Health Care domain: cognitive and physical-motor therapeutic intervention in water and, more specifically, from the observation of the therapists working at the swimming pool with disabled children, their practices, their needs and objectives.

The hypothesis behind the research is that *palpable computing technologies can offer a range of functionalities that directly address the needs of impaired children and their caregivers.*

In particular, the Active Surfaces design accounts for the need for configurability, constructability, modularity, physicality and support for creativity in the therapeutic practice. It is also an aid for engaging the children in playful activities by supporting movement, manipulation and cognitive involvement.

This vision was guaranteed by continuous dialectical relationship between use and development of technology throughout the process. Continuous iterations between trials and evaluations with the users and fine-tuning of the prototypes took place. The therapists have been involved in every phase of the research, as they are part of a participatory design process. They participated in the design of the Active Surfaces from their inspiration to development provided design inputs, critical feedback and their evaluation. Such mutual exchanges resulted in a richly fertile dialogue through the whole research process.

Palpable Computing couldn't be described without users participation essentially because it envisions ubiquitous technologies as becoming clearly available throughout actual and potential users activities. Palpable use of systems can be summarized as the opportunity people have to understand what is going on at a level they choose and to support the user's control and choice.

In our case study we are interested in the specific application of Ubiquitous Computing technology to the educational and therapeutic domains for children with special needs. In these domains, technology should contribute as an aid to human thought and reflection that helps people become self-aware, envision personal and learning goals, and experiment with creative options for making progress (Sharry et al. 2003).

The Ubiquitous technology used in therapeutic play with disabled children may prove to be compelling and engaging to children, motivating them to become curious about how they act in the material environment and to try out new skills in collaborative and competitive activities.

The technology, however, only provides support for therapists and special education teachers and is not a replacement for a quality therapeutic activity. The essential therapeutic element is not the technology itself but the dialogue between children and caregivers, which is often based on physical contact rather than words, and which might be powerfully improved by the physical arrangement of ubiquitous and palpable computational resources.

Technologies that are specifically designed for people with special needs have an undeniable social relevance and can be important within a wider research context since they contribute to understanding how the actual capacities and cognitive abilities may be augmented, whatever their strengths and weaknesses might be. In order to reach this goal, it is worthwhile to account for the heterogeneous special needs of disabled users. People obviously don't share a common and unique profile; rather there is great variability among all the possible different users regarding personal attitude and sensitiveness, and physical and cognitive abilities. In this research we study how the use of Ubiquitous Computing may increase physical activity, encourage greater motivation, a longer attention span and engagement. This is also achieved through the combination of computer science and social science methods, so that new technologies are solidly anchored in actual work problems and practices. This has resulted in software and hardware implementation solutions for therapeutic activity in the water.

Organization of the thesis

This approach is also reflected through the structure of the thesis. After this introduction we give an overview of the Conceptual and Architectural framework of the research with a particular focus on Architectural Qualities and Use practices (Chapter 1). In this chapter the research theme and the methodological approach will be specified. The role that Ubiquitous Computing might play in designing for user diversity will also be described both at the conceptual and at the methodological levels.

The fieldwork at the swimming pool is then described in extensive detail (Chapter 2). The activity analysis and modelling will also be illustrated in order to gain a firm understanding of the therapeutic practice in

water for children with special needs. Theoretical approaches, enabling technologies and situational factors will be investigated.

The design and development of the Active Surfaces application prototype will be described (Chapter 3). The concept development and evaluation will also be documented.

The Architectural Qualities will be discussed with relation to Active Surfaces (Chapter 4). A description of the PalCom open architecture will be given and the architectural and non- architectural prototypes will also be detailed.

The experimentation with the Active Surfaces prototypes will be described in great detail (Chapter 5). Performance testing and User testing will be described through methods, tasks and trials, and discussion of results. The discussion of the experimentation will concentrate on Architectural Qualities and Use Practices.

Conclusive discussions will be addressed in the last chapter of the thesis (Chapter 6). Main research themes of the thesis will be discussed and open questions will be outlined.

The experience described in this thesis has been a valuable opportunity for me to explore interesting research themes, like architecture for Ubiquitous Computing, therapeutic technology, the exploration of the water environment, and user diversity. I had the opportunity to take part in a multidisciplinary project like PalCom and work in an interdisciplinary group within which diverse backgrounds co-exist and integrate in a productive way, i.e. there has been a combination of competences in HCI, interaction design, psychology and software engineering. This resulted in the integration within this thesis of different methods and a productive dialogue among user studies, design and software development. The work was conducted by the Interaction Design Unit at the Communication Science Dept. of the University of Siena and, as a part of it, the author took part in the whole research process and most of the reported activities.

Because it is part of an EU funded IP that lasted four years (Jan 2004- Dec 2007), we must clarify that the research described in this thesis is part of a bigger project in which more than a hundred people have been involved. Twelve institutions, universities and companies were part of a heterogeneous project consortium in which backgrounds in computer science and user studies met and integrated. Thus this thesis couldn't be possible without the joint collaboration among the Interaction Design Area, The University of Siena, and the Computer Science Dpt., The University of Aarhus, the Aarhus School of Architecture, the Computer Science Dpt. and the Lund Institute of Technology. The contribution of each partner in the research will be specified throughout the thesis.

The specific contribution of my PhD consists in field study, concept development and evaluation, a definition of requirements for architectural development, the design and development of the Lightweight embedded prototype, and the experimentations with Active Surfaces, including both the Performance testing in the laboratory and the User testing in the swimming pool.

Chapter 1

Architectural Qualities and Use Practices

1.1 Architectural qualities for palpability

For the last twenty years, the design and development of software architectures for interactive systems have been a major concern in academic research and industry (John, Bass 2001), and *real use* and *usability* have also become issues of research interest. Usability benefits have been widely applied to individuals performing on a desktop computer but need now to be re-examined within the context of distributed, interactive, networked and embedded applications. Usability studies, which traditionally approach aspects specific to a given task or application, have to be reinterpreted and adapted to Ubiquitous Computing application systems, wherein networks of laptops, PDAs, wearable computers, mobiles and other distributed devices are integrated.

In fact, as described in the introduction, the fields of ambient, ubiquitous and pervasive computing, following the inspiration of Weiser (1991), extract computing and communication resources out of standard desktop machines and embed them in distributed, small and multipurpose devices within the environment.

The Ubiquitous Computing early vision (Weiser, 1991, 1993) proposed a richly renewed environment in which people deal with a multiplicity of networked communication and processing devices. This moved the HCI paradigm toward a more dynamic perspective within computers that were widely spread throughout the physical world.

Designers and developers must also find ways in which sensitive, responsive and intelligent UbiComp technology can also become *usable*, i.e. noticeable, comprehensible, adaptable and easy to control. That is why usage and usability concerns need to be reconsidered outside of the desktop metaphor. Achieving usability traditionally depended on how the functions provided by the system were understandable and clearly visible through the user interface. In this paradigm users have many input and output peripheral devices and the overall system *interface* must be adequate for their needs.

There is a multitude of interfaces and usability issues for each mobile device of the distributed and ubiquitous system, and this requires a unique and enabling software architecture that must be designed according to users' needs. Making ambient and ubiquitous computing *palpable* takes computing to the interoperability, coordination and presence of multiple and different devices, each of which must rely upon the same middleware architecture.

Middleware is “the software layer that lies between the operating system and the applications on each site of the system” (ObjectWeb Website). Middleware is intended to make it easier for the software developers to create applications that satisfy specific criteria so that they can rely on primitives provided by the middleware layer.

The use of a service-oriented architecture (SOA) has also been widely used in the last few years as the architectural approach for various types of ubiquitous, ambient and pervasive systems (O'Brien et al. 2005). ‘Service-oriented’ architecture can be defined as an architectural approach to build systems or applications that use a set of services. This is one of the means of modularizing the software so that modifications may be fast and easy. Infrastructure for Ambient and Ubiquitous Computing is an emerging field (Edwards et al. 2003; Bellotti et al. 2002), in which flexible and powerful architectural mechanisms are required. Architectural support is needed for designing embedded, distributed, intelligent and interactive systems, which have scalability as a special feature. Much of the theoretical challenges described in the introduction of this work need to be addressed, explored and experimented with by using a common enabling architecture. In particular, together with scalability, adaptability, heterogeneity, global access, and the quality of service in dynamic and ad-hoc environments, the additional requirements of palpable systems are:

- support for the construction and de-construction of both the architecture and components running on the architecture,
- availability and dynamicity of components and structures as needed,

- balancing reactive behaviour (e.g., with the user in control) and proactive behaviour (with the system in control) (PalCom Project Website).

Ubiquitous applications should support highly mobile environments with numerous multi-sized and distributed devices; consequently, the middleware needs to adapt to these requirements.

In contrast to the programming of the large (PitL) approach (DeRemer and Kron, 1976), which is focused on the development of large-scale software systems, this research focuses on *programming in the small and many*, similarly to the Prism Project (Medvidovic et al. 2003). The PalCom middleware architecture has also been developed for distributed, dynamic, mobile, heterogeneous computation in small, resource-constrained devices. Most of the technology is realized with embedded systems or contains embedded systems. Devices such as actuators, sensors, microprocessors, displays, RFID tags, etc. will be embedded in the environment, just as mobile smart phones, PDAs and mixed-media devices contain embedded systems. Architecture for UbiComp should allow the experimentation in practice with the idea of *run everywhere*, which was originally adopted by the Java marketing division and allows users to make sense of the computation present in the material environment in which they live. That is why the open architecture and palpable runtime developed is expected to support devices with a small amount of RAM (lower bound in the order of 256Kb) and slow, low-powered microprocessors (PalCom Project Website, PalCom Deliverable 22).

This work attempts to understand the architectural mechanisms required by usable ubiquitous technology that can be explored in resource-constrained devices. We try to bridge the system architecture to specific aspects of use, e.g. the ability for a user to operate specific functions and their implications for software infrastructure. This would clarify the specific coupling between use practices and the corresponding enabling architecture mechanisms, and it is also crucial to understanding the potential the system may offer in possible future applications.

In this research we identified a set of specific use opportunities that have software architectural implications, especially when designing for palpable Ubiquitous Computing systems. These uses aimed at being exhaustive with relation to the activity taken up in the case study considered, and they cover aspects that contribute to attaining the *palpable use of ubiquitous technology*.

In the context of performing and evaluating software architecture the selection of the overall system structure is thought to provide software quality through attributes like modifiability and performance (John, Bass 2001).

This is achieved through design decisions embodied in the software architecture and these decisions may also preclude the implementation of a fully usable system.

Through these decisions developers may provide novel use opportunities and remove constraints. Some uses we are normally dependent upon while using desktop machines, such as “cancel” and “undo”, require software architecture support (Bass, John 2002). Because they reach so deeply into the architecture of a system, these facets must be built into the system from the early development rather than added after the initial system design. The research framework often provides the main architectural and theoretical concepts that suggest opportunities that users may want to be provided with. Other exemplar usability features are Using Applications Concurrently, Recovering from Failure, Reusing Information, Working at the User’s Pace and Predicting Task Duration (Bass et al. 2001). Architectural decisions such as these are critical to change throughout the design process because they are costly and time consuming (Bass et al. 2003).

Attributes such as those hereby discussed referred to software architecture *quality* and *usability*. In this research we focus on those attributes that support *palpable use* of technology, that we henceforward call *Qualities*.

The concept of *palpability*, originally inspired by and developed through empirical ethnographic and ethno-methodological studies with professionals, has subsequently been addressed within the design and development of the software architecture. In this process, the palpability of ambient and ubiquitous systems has been translated into *architectural attributes* and the *Qualities* to guide development and experimentation.

The Qualities are inspired by:

- The initial conceptual framework and the formulation of the challenges, see the Introduction.
- The continuous iterative development of the general architecture for palpable computing and the specific application within the Active Surfaces application prototype.
- Internationally standardized definitions of architectural quality attributes, such as those identified in the ISO/IEC standard 9126 dealing with product quality in software engineering (PalCom Deliverable 54).

They predominantly describe fundamental *non-functional* attributes (Bass et al. 2003) of the PalCom architecture, although they can also include aspects of functional behaviour.

The Qualities are specific to the PalCom architectural design posited for palpability. They individually describe separate facets, and collectively describe the emergence of palpability as a property-in-use of computing made palpable.

Each Quality is described in the following section. The resources we refer to in each description are the fundamental entities present in any palpable system (PalCom Deliverable 54). The *computational resources* we refer to in the rest of the thesis are software programs; the *physical resources* are devices and the basic resources associated with them, e.g. memory; and the *human resources* are the actors who interact with computational and physical resources.

The architectural Qualities of the PalCom open architecture are defined as follows (PalCom Deliverable 54):

- **Resource Awareness:** Computational, physical or human resources are aware of one another's presence, availability and behaviour.
- **Assemblability:** Manual or automatic assembly, and disassembly, of resources into composite constructs that exhibit the collective sum behaviour of the constituent elements, or, in some cases, emergent behaviour that is only brought about through the collaboration of particular resources.
- **Inspectability:** The attributes and behavioural capabilities of a resource can be exposed at various degrees of visibility using monitoring and interrogation. Monitoring passively observes behaviour, and interrogation actively probes behaviour through available inspection interfaces and constructs.
- **Adaptability:** Certain resources can dynamically change their behaviour in response to detected events or environmental conditions.
- **Resilience:** When required palpable systems can exhibit behaviour that ensures a defined degree of reliability and survivability. Reliability implies that a system will attempt to do what is expected at all times, adapting its behaviour as necessary to ensure such an outcome. Survivability builds on reliability to ensure that a system can tolerate problem conditions and continue to perform as expected even in the presence of disruptive events.
- **Multiplicity:** A palpable system of computation is formed through interaction between resources, whether cooperatively or competitively. Any given resource may be participating in multiple simultaneous, dependent or independent, interactive relationships.
- **Experimentability:** Systems conforming to palpable computing principles facilitate and encourage experimentation by users of those systems, because palpability intrinsically relates to the ability of a coherent collection of resources to be used, customized and altered within established degrees of freedom and constraints, such as performance and security.

The purpose of the Qualities is to capture the essence of what defines the nature of palpability in ubiquitous computing. The Qualities link the conceptual aspects of palpable computing to its architectural realisation. Moreover they should not be considered in isolation, but rather as interwoven contributory factors that exhibit dependencies and influences on one another. In fact, for a system to be considered palpable, it must exhibit all of these qualities to a degree appropriate to the application context.

In this thesis, we will explore in depth part of the qualities introduced above. In fact, although all of them are present at different levels and emerge through the interaction, four architectural Qualities are specifically realized by the Active Surfaces application prototype:

- *Assemblability*
- *Adaptability*
- *Resource awareness*
- *Experimentability*

These qualities are more thoroughly addressed and discussed in the following paragraphs in terms of their relationships with a resource, or set of resources, which are part of the software architecture that is fundamental for making computing palpable.

Assemblability

Assemblability is inspired by the modifiability quality attribute of Bass (Bass et al. 2003) and it is strictly related to the challenge that palpable systems should complement construction of systems with deconstruction and reconstruction. Assemblability describes the fact that resources can be assembled into multiple, composite constructs that exhibit the collective sum behaviour of the constituent elements, or, in some cases, emergent behaviour that is only brought about through the collaboration of certain resources.

As part of the early vision of UbiComp (Weiser 1993), the ensembles or assemblies are indicated for the powerful emergent capacities that arise from the complex interconnection of devices and services in ambient computing environments.

Instead of embedding computing everywhere in the environment it considers how UbiComp technologies can be created as an assembly of resources, which can be mobile and/or fixed, to serve specific purposes and be situated in particular places. Furthermore, it argues that people, rather than computers, should take the initiative to be constructive, creative and, ultimately, in control of their interactions with the world in novel and extensive ways.

Assemblability provides the construction and deconstruction of services, components and devices that fits into a common model. It provides flexible interfaces and allows for dynamic composition and decomposition.

Assemblability describes the capability that resources have to be assembled into multiple composite constructs that exhibit the collective sum behaviour of the constituent elements, or, in some cases, emergent behaviour that is only brought about through the collaboration of particular resources. Moreover, any assembled construct must be equally able to be dynamically disassembled or reassembled into alternate formations in real time

People create richer information out of their experiences, by integrating and putting different resources together in a creative fashion. In socio-technical 'systems' comprised of human actors, technologies and materials, the creation of stable and effective practices involves continuous bricolage (Shapiro et al. 1996): the often ad-hoc and creative combination of the materials at hand for a particular purpose (PalCom Deliverable 12). By changing and experimenting with different elements we may provoke unknown outcomes and solutions and expand the possibilities of the artifacts.

Assemblability is strongly related to adaptability, which describes the fact that any assembled construct can be disassembled or reassembled into alternate formations in real-time. It then ties in with adaptability by allowing construction, deconstruction and reconstruction to be carried out manually or automatically.

The provision of this quality is influenced by the quality of experimentability, in that an experimental mode of

interaction must be supported for the manual handling of multiple composite constructs.

Adaptability

A system exhibits adaptability if it is possible to change the behaviour of the system by replacing, or altering the behaviour of, some of its elements without affecting the rest of the system. Adaptability describes the fact that certain resources can dynamically change their behaviour in response to detected events or environmental conditions.

The main difference between our definition and the original from Bass (Bass et al. 2003) is that we consider systems that are either running, or being designed (Bardram et al. 2005). Computational resources are brought into a functional aggregate, the composition of which can vary dynamically. Dynamic resource reconfiguration and system behaviour modification can be effected by either programmatic autonomous means or through human interaction. Due to the changing conditions of the environment and resource availability, users need to continuously understand and adapt to the environment by navigating, questioning and modifying the available information resources. The assembled resources can at any time be modified by manual changes: the users manage and adapt the system configuration flexibly to their needs. Parts of the assembly can be replaced dynamically with others available in the operating environment. In this way emergent and un-anticipated usages of resources are predicted and supported.

Assemblies can also be altered with a degree of self-configuration due to the possibly high number of devices connected to each other. This is probably beyond the limit of manual configuration, and the mechanisms also allow for split device solutions, where functionality comes from the composition of different entities, each contributing its share to achieve the common goal (PalCom Deliverable 54).

Resource awareness and inspectability provide the necessary programmatic abstractions to enable human or algorithmic reasoning about adaptations, while assemblability provides the means to make it effective.

Resource Awareness

Hardware and software resources are brought into a functional aggregate whose composition can vary dynamically. Due to the changing conditions of the environment and resource availability, users need to continuously understand and adapt to the current status by relying on the possibility to navigate, question and modify the system registry (included in the Hierarchical maps describing the system components).

Resource awareness describes the fact that resources are aware of one another's presence, availability and behaviour. Being aware of the presence of a resource implies the ability of one resource to discover other resources. In turn, this implies that discoverable resources are able to announce their presence, or to have their presence announced by others. Being aware of the availability of a resource implies the ability of one resource to know whether other resources can be used or manipulated. Being aware of the behaviour of a resource implies the ability of one resource to determine the operational capabilities of other resources (PalCom Deliverable 54).

Parts of the assembly can be replaced dynamically with others available in the operating environment. The assembly can be changed in terms of its constituent elements by decomposing those no longer needed and/ or replacing them with others. This change is based on user's actions, mostly unbinding and replacing resources.

Experimentability

Experimentability describes the capability of facilitating and encouraging exploratory experimentation by users participating in systems that conform to palpable computing principles. This is because palpability intrinsically implies the ability of a coherent collection of resources to be used, customised and altered within established degrees of freedom and constraints, such as performance and security (PalCom Deliverable 54).

Experimentability relies on resource awareness and inspection to provide an overview of what can be experimented with, and on assemblability and adaptability to enable an intervention to be made when required. Experimentability

should also ensure that users be able to trust that exploratory experimentation will not yield destructive results.

The Qualities, as attributes referred to palpability, define how the architecture contributes to a decrease in the complexity of building, maintaining, and understanding UbiComp systems. Users thus become actors of the computing infrastructure that is made use of in the application prototypes.

In the PalCom project the software architecture has been explored and experimented in different general scenarios and application prototypes related to the Health Care and the Landscape Architecture domains. Each general scenario is characterized by an application prototype in which the PalCom architecture, or part of it, has been experimented. The application prototypes served as testbeds for the development of the architecture and as case studies that inform the development by providing the requirements coming from the field studies.

The qualities that define the PalCom architecture are informed both by the use of the application prototypes in the real contexts and by the development of the architecture.

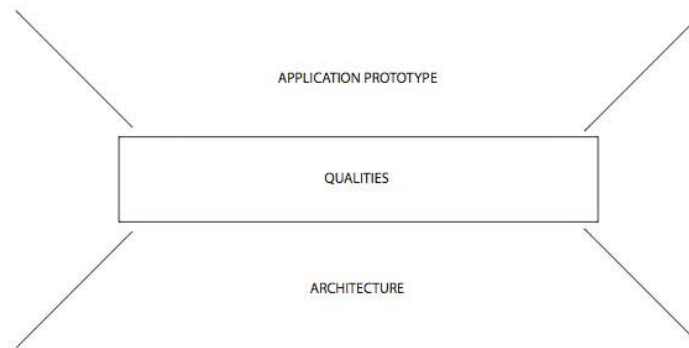


Figure 3. Qualities between Software Architecture and Application Prototype

The software's architectural Qualities represent key research issues addressed by this thesis. In particular the qualities are part of a process that:

- underpins the conceptual framework of PalCom (PalCom Deliverable 37),
- directly informs the design of application prototypes (PalCom Deliverable 50),
- informs the formulation of scenarios (Deliverable 17),
- informs the design of the PalCom architecture.

The Qualities for palpability are not strictly informative on the software *quality* of architectures engineering. The ISO/IEC definitions (Bass et al. 2003) provide a background standard on top of which the non-standard qualities of the Palpable Computing architecture have been built.

1.2 User Diversity and Use Practices

The PalCom architectural qualities have been introduced and described as the meeting point between architecture and use/application. The peculiarity of these non-standard architectural qualities is that they both evolve and gain meaning through software development and investigation of key *use practices* to account for.

In fact the *architectural qualities* and the *use practices* are tightly coupled and represent the two perspectives adopted in this research: the software architecture engineering and the interaction design perspective. In order to better focus on use practices as they emerge in the Active Surface application prototype it is thus necessary to observe and describe in depth the target user groups - that is, the therapists and caregivers together with the disabled

children - their needs, wishes and abilities (Pollini, Grönvall 2006). The fieldwork and activity analysis described in Chapter 2 will specifically address the therapeutic *Use Practices* within the Active Surfaces case study (Pollini et al. 2006).



Figure 4. Playing domino like games with Active Surfaces

We start with discussing the peculiarity of designing for users with special needs and diverse abilities, and how the opportunities offered by the Ubiquitous Computing technology may address special needs and wishes by providing flexibility and extended scope (Rogers 2006).

The push towards developing assistive UbiComp systems is especially directed toward the monitoring and sensing the elderly and the physically and mentally disabled in homecare (Mynatt et al. 2004). In many of the current UbiComp applications the movements, habits and health of disabled people are recorded, tracked and presented to the families, caregivers and other individuals responsible for them via remote monitors. A number of assisted living applications and services have also been developed to help people with loss of vision or deteriorating memory to be more independent in their lives (Rogers 2006; Tran et al. 2005). For example, Cyber Crumbs was designed to help people with progressive vision loss, promoting orientation by using a reader badge system (Ross 2004).

Performing Ubiquitous Computing technologies have profound consequences on the Use Practices with regard to the type, the content, and the functionality of the devices and services (Emiliani, Stephanidis 2005). It provides novel and emerging opportunities that impact a wide variety of human activities.

Scalability, as one of the key issues in Ubiquitous Computing, provides the technical condition for enabling novel use opportunities. In fact, UbiComp is characterized by applications that address scalability in terms of space, people and devices. Eventually, computational resources will need to be placed in different settings, so they can range from a one inch-scale (PDAs, Voice Recorders, smart phones, etc.), to a foot long-scale (notebooks, tablets, digital paper, etc.), to a yard-scale (electronic whiteboards, plasma displays, etc.).

The present research contributes to this discussion by experimenting with the UbiComp architecture developed in PalCom in microprocessors. The PalCom architecture, on a scaled down level, provides opportunities for interactive and distributed handheld micro-devices and enables *flexibility and extended/ specific scope* for applications (i.e. play and therapy) with less able users.

The PalCom architecture enables the performance of Ubiquitous Computing to address a variety of issues that are critical for elderly and disabled people. In particular, PalCom technology may fulfil the requirements given for universal access and user diversity (Emiliani, Stephanidis 2005). The architecture can be used to develop applications that are:

- personalized and tailored to user needs: i.e., it can recognize the user, evolve throughout time and keep track of the history of the interactions,
- adaptive: i.e., its behaviour can change in response to a person's actions and environment;

- anticipatory: i.e., it anticipates a person's desires and environment as much as possible without mediation.

UbiComp devices are no longer perceived as computers, but rather as augmented elements of the physical environment. They are equipped with built-in facilities for multimode interaction and alternative I/O (e.g., voice recognition and synthesis, pointing devices, vibration alerting), addressing a wider range of user and context requirements than the traditional desktop computer.

It is important when ensuring universal access to these technologies (in particular, for elderly people and people who are disabled), to investigate how people will be engaged in the emerging forms of interaction, and how this will affect their physical, emotional, psychological and cognitive experience. The main challenge users provide is to identify and avoid forms of interaction that may lead to negative consequences, such as confusion, cognitive overload, and frustration (Emiliani, Stephanidis 2005).

In this research study we deal with an application that realizes some aspects of Ubiquitous Computing in the scenario with users with special needs. By mean of this application we can also explore real experiences of motivation, engagement, coherence and reward, as well as frustration, hyper-stimulation, cognitive overloads and confusion.

Because they stem from accurate user studies, the developments in UbiComp might also impact the field of assistive technologies. Assistive Technology is defined as "any product, instrument, equipment or technical system used by a disabled or elderly person aimed to prevent, compensate, relieve or neutralise the deficiency, the inability or the handicap" (International Standard Organization ISO-9999). In fact, UbiComp technology, as is also possible with the PalCom architecture, enables a shift from proactive technology (Weiser 1993) to proactive people (Rogers 2006). Palpable UbiComp technologies are designed for engaging Use Practices where technologies are designed not to do things for people but to engage them more actively in what they currently do and want to do. In contrast to pure assistive technology, Ubiquitous Computing that has been made palpable for users focuses on users that proactively adopt computational resources in a responsible and purposeful way.

Some of the main challenges of Palpable Computing (described at the beginning of the thesis) specifically address key issues in the universal access perspective (Emiliani, Stephanidis 2005), they are:

- automation and direct control,
- the identification of dependencies among constructed and de-constructed devices and services,
- dynamic adaptation over time.

The effective control of dynamic and distributed systems also becomes critical, specially in order to establish an appropriate balance between automated and intelligent processing, and human intervention aimed at directing and modifying the system. People are allowed to take the initiative to be constructive, creative and in control of their activities.

Then the interest is on the manners (*how to*) UbiComp technologies can be created as *assemblies* (*ensembles* or *ecologies*, in Rogers 2006) of mobile and fixed resources, in order to support the enhancement of health, therapy and play practices. Users can construct their own applications by manipulating both embedded devices and software services useful to set the functionalities.

The assemblies created have then their own existence throughout time and users need to flexibly adapt them to eventual dynamic changes.

The key Use Practices treated in this thesis are outlined below. They will be specified and described in detail within the Active Surfaces case study in Chapter 2.

The PalCom architecture allows the development of technology that can be constructed and combined as unexpected for purposeful interactions (*Assemblability*). In all the application sites people combine different technologies, materials, objects and artifacts. The assemblies of tools, that can be planned or ad hoc, bring together diverse elements and the newly created formation is intelligible as a whole. By changing only one single element,

we may provoke a process of *bricolage* with novel outcomes (PalCom Deliverable 12).

These assemblies are embedded in a socio-technical ‘system’ of human practices, other technologies, and materials. The creation of stable and efficient tools within such socio-technical system is fraught with uncertainties. The dynamic nature of the context involves continuous adaptation of the systems (*Adaptability*) either based on user decision or as autonomous behaviour. People usually need to re-configure the resources in daily practices in respect to their objectives, the changing conditions of the environment and the behaviour of the other actors.

Throughout their activity people often dynamically move from context to context, constructing, deconstructing assemblies of technologies, objects, materials, and artefacts, depending on the actual and specific needs (PalCom Deliverable 12). Many different systems and applications may be involved, and information about available resources may move in and out of digital and physical formats. *Resource awareness* provides the ability of finding and discovering the available opportunities for action and to control the state of each component, capture possible faults, anticipate failures and prevent system breakdown. In a dynamic context people also adopt the tools in creative ways and try the newly created solutions out. They flexibly handle the available resources (*Experimentability*) without any specific regard to the predefined behaviours and the proper performance of the systems.

1.3 Research methodology

Such investigation requires novel design processes and methodologies that are appropriate for Ubiquitous Computing. Furthermore, dealing with diverse and special users requires that methods and experimental environments would be non-obtrusive, able to be personalized, adaptable, and capable of anticipating emerging user needs (Emiliani, Stephanidis 2005). Requirements for therapeutic environments need to be developed, tested, and evaluated with the stakeholders.

Toward this goal, a core set of PalCom Architecture components is used and implemented in a number of different architectural prototypes. The PalCom architecture offers the possibility of experimenting with use solutions from different perspectives (disabled people and caregivers) by providing a common middleware environment for solutions suitable to special needs.

A wide variety of methods have been used throughout the iterative design life cycle (further described in Chapter 3). These methods pertain to Human Computer Interaction, Participatory Design and Software Architecture Engineering. We have experienced with integration among the traditional ethnographic studies, participatory design methods and naturalistic experiments to inspire, inform and evaluate the design of software architectures (Bardram et al. 2004). This approach has already been adopted for the design of ubiquitous computing technologies (Stringer et al. 2005; Diggins, Tolmie, 2003) while it seems to be still fully appreciated in software architecture design (Edwards et al., 2003).

In this study we integrate a participatory design perspective with a co-evolutionary approach to interaction design and we explored this methodology in the domain of software architecture design. The process, as part of the PalCom project, is co-evolutionary since architectural development, site exploration, activity analysis and concept design have been carried out in parallel so that each path of the process can inform, without constraining, the others. In fact the PalCom architecture has been experimented throughout the whole process in a number of application prototypes, among which the Active Surfaces, which have allowed continuous dialogue with the users and in-depth investigation of use practices.

User driven and *design driven* methods have been combined. The convergence and the iterative exchanges between user work and technology development is realized through a variety of tools, i.e. the architectural and application prototypes and the different kind of scenarios that have been developed. The experimentation and evaluation with

the users have been performed to compare and identify strengths and weaknesses in architecture developments. In fact the existing systems have been experimented within real contexts of use. The field experiments helped software developers make sure that the architecture would be able to fulfill the requirements of the key Use Practices addressed and will reify the architectural Qualities for making Ubiquitous Computing palpable.

1.3.1 Methods

We try here to specify the relationships between the different methods followed in this research and describe how these have been integrated in order to produce relevant and informative outcomes. These are:

- ethnographic methods
- participatory design
- scenario-based design
- architectural prototyping
- simulation-based methods

The participatory approach to system design has been enriched over the years by ethnographic methods of analysis and it has been directly oriented toward the support of the practices of the users rather than the automation of their work. The ethnographic observations figure in the architectural characteristics of ubiquitous computing (Tolmie et al., 2002) and context aware systems (Bellotti, Edwards 2001) and appear to be necessary to understand the practice of using a multiplicity of distributed networked devices. Together with it, participatory design plays also a relevant role in our research process. This tradition is based on the assumption that it is the users who will best be able to express what kinds of functionality are needed, and that their democratic participation (Ehn, Kyng 1987; Ehn, 1988) in the design process will put real demands on the design practice. In particular, the role of the users in the design of UbiComp technology is to actively take part in an interdisciplinary design team by providing the knowledge on the field that can be transformed into design solutions.

The use of scenarios (Carroll, 1999; 1995) helped the structuring of data gathered through activity analysis, the envisioning of the role and functionalities of the system, and the assessing and validating the envisioned solutions from an architectural perspective. In this way scenarios work as a design tool along the overall design process.

Scenarios represent one of the most valuable techniques for representing current activities and analysing and planning how a new system could impact on users' activities and experiences. A scenario identifies people having certain motivations toward the system, describes the actions taken and some reason why these actions were taken, and characterizes the results in terms of the users' motivation and expectations. They project a concrete description of the activity the users engage in when performing tasks, a description sufficiently detailed so that the design implications can be inferred and reasoned about. Using scenarios in system development helps keep the future use of the envisioned system in view as the system is designed and implemented (PalCom Deliverable 16).

From a software architecture design perspective the scenario-based method is a mean to evaluate particular attributes of the architecture. In this research we refer to the Qualities as attributes for palpability as experienced by the users. The design and development of the architecture has been supported through the creation of scenarios that forces a very concrete description of the Qualities.

Several scenario-based evaluation methods have been used in literature for software architectures, e.g., the Software Architecture Analysis Method (SAAM) (Kazman et al. 1994), the Architecture Tradeoff Analysis Method (ATAM) (Kazman et al. 1999), and the Architecture Level Modifiability Analysis (ALMA) (Bengtsson 2002).

Throughout the whole process the scenarios are used to step through the software architecture and to document the consequences of architectural solutions from a user perspective.

At the beginning of the project, we generate a number of scenarios partly elaborating on user observation and work analysis, partly managing in parallel an open phase of creative concept generation. The scenarios specify in a narrative form the features of the PalCom architecture in relation to the user activity, interaction paradigms, information contents, and preliminary technological solutions.

Different kinds of scenarios drove the research process: Activity scenarios, Envisioning scenarios, Prototype scenarios and Qualities scenarios. These categories of scenarios are described in the following section and will be further illustrated in the following sections. The descriptions of the scenarios are also part of the PalCom Deliverable 16.

Activity scenarios stem from the fieldwork and activity analysis. They are grounded and built on data collected with ethnographic observation and story telling through focus groups, interviews, diaries, cultural probes. Activity scenarios account for concrete use episodes, from standard practices to routines, from routine activities to exceptional circumstances faced in daily activities.

Envisioning scenarios represent a tool to envision the future system, a first way to embody ideas and explore the possibilities to support these ideas. Envisioning scenarios stem from the inspiration phase of the design process. They support the construction of a coherent and rich vision of the concept and provide support for proof-of-concept. Envisioning scenarios fix some of the narrative elements and use them as leverage for creative generation of design solutions.

Prototype Scenarios. According to the approach proposed by Houde and Hill (1997), interactive prototypes represent the evolution of mock-ups, integrating *role, look and feel* with *implementation* aspects. Scenarios for prototype evaluation represent detailed interaction paths, and have the function to allow the refining of the interaction modalities and the sensorial features of the prototypes. Scenarios for prototype evaluation must enable a clear grasp of the contextual elements, and have to afford specific requirements coming from the activity.

Qualities scenarios. They consist of a slight adaptation of the quality attribute scenarios (Bass et al. 2003). The specific Qualities for palpability are defined ad-hoc within the conceptual and architectural framework of the PalCom project (PalCom Deliverable 54) and the Qualities scenarios are a way to make these palpability-attributes operational. Qualities scenarios provide both a context for a Quality as well as a way to concretely measure whether the architecture fulfils the requirements of the scenario. It states measurable properties of an architecture by defining metrics to be used in performance testing of the architecture. They are expressed using a standard template having six parts:

- the source of stimulus,
- the stimulus itself,
- the artifact that is stimulated,
- the environment that expresses the condition under which the stimulation of the artifact happens,
- the response that the stimulus generates,
- the response measure that defines a metric to measure the response.

The development of different scenarios is the means by which we can gain insights into how selected activities could be mediated by current and future Palpable Computing technology. The aim of this activity is to create a set of scenarios used to:

- analyse the role of actors, spaces, tools and objects in Use Practices (activity scenarios);
- envision how the technology will shape and transform the activity by the introduction of innovative design concepts and the use of working prototypes (envisioning and prototype scenarios)

- experiment with and evaluate specific features of the technology by testing attributes and Qualities of the enabling software architecture (qualities scenarios).

The scenarios that have been developed strictly focus on the Architectural Qualities and the Use Practices (Par.1.1 and 1.2) and are used to integrate contributions coming both from the application/use side and the architectural side. These two levels of analysis are both present in scenarios; in some cases the former is more emphasized (i.e. the Activity scenarios), in other cases the latter (the Prototype scenarios and the Qualities attributes scenarios). Each scenario would deal with specific aspects of the Qualities.

The Activity scenarios will be articulated according to use issues coming from fieldwork and activity analysis, specifically the objectives and the users (Par. 2.1), the tools, the phases of the activity and the practice of the therapists (Par. 2.2). The activity scenarios elucidate requirements for the design of the technologies and the enabling middleware architecture. Activity scenarios will be described in Chapter 2.

The Envisioning and Prototype scenarios will then describe the introduction and the impact of the Active Surfaces concept and prototypes on the practice. Those scenarios are elaborated on the Activity scenarios and are used to evaluate the design intuitions and solutions with the users. Envisioning and Prototype scenarios are described in Chapter 3 and Chapter 4.

The Qualities scenarios have a more technical nature than the previous scenarios. The Qualities scenarios attempt to describe the behavior of a system and to provide a way of measuring it in an actual context of use. Qualities attributes scenarios, their relation with the existing literature and their use in performance testings, are described in Chapter 5.

Several methods have then been used for experimenting and evaluating the PalCom software architecture. The extensive use of architectural prototypes is one of the main important. As an overall classification we can describe traditional software prototypes according to the following characteristics (Christensen 2003):

Horizontal vs. vertical prototypes:

- Horizontal prototypes focus on what is visual in the user interface, and the underlying functionality is often “hacked”, simulated or even left out.
- Vertical prototypes have selected areas where the functionality is implemented in (almost) the detail of the final system.

Exploratory, experimental and evolutionary prototyping:

- Exploratory prototyping, the prototype is used as a means of requirements gathering,
- Experimental prototyping, the prototype is used to determine whether an already perceived idea is adequate,
- Evolutionary prototyping, the prototype is gradually developed into the final system either in a incremental way in which more and more is added stepwise or by iteratively refining what has already been covered.

More specifically the architectural prototypes are means used for designing, building, and evaluating software architectures (Bardram et al. 2005) and consist of a set of executables created to investigate architectural qualities, such as those we have described in Par. 1.1. The architectural prototyping is a technique for exploring the architectural design space and for addressing issues regarding Qualities specific attributes for the target system.

Following the classes proposed by Floyd (1984) a similar classification of architectural prototypes as either exploratory or experimental have been made (Bardram et al. 2005):

- the exploratory prototypes are built to explore alternative architectural solutions and clarify requirements
- the experimental prototypes are used to gauge the adequacy of a proposed architecture, typically by directly measuring or analyzing software architecture qualities.

In this research *experimental architectural prototypes* have been used to conduct experiment on the architectural qualities that we have analyzed, in particular those observable at run-time (like performance) (Bardram et al. 2005).

The experimental architectural prototypes allow concrete measurements to be made under a range of different situations and these might be also defined in terms of short technical scenarios referred to specific Qualities.

Another main method of our research approach is inspired by the *simulation-based method* in software engineering (Bosch 2000). The Simulation-based method relies on a high level implementation of some or all of the components in the software architecture and its environment. The simulation has been used to evaluate the architectural Qualities and related specific application requirements. In this study the simulation has also been combined with prototyping: *embedded* and *simulated* experimental prototypes of the architecture are developed and experimented in the intended context.

Together with the embedded and simulated architectural prototyping, a *lightweight embedded prototype* has been developed. In spite of being a non-architectural specific technology, this prototype expresses some of the core objectives of the infrastructure. It leverages the fullest extent of the power of the infrastructure, given the assumption that this is the critical thing to test first (Edwards et al. 2003).

It has been used as proof-of-concept application that succeeds in demonstrating compelling new user experiences. The empirical observation of real use of the lightweight prototype was essential in proving the application's significance, still not probing early the realistic use of the final system (Edwards et al. 2003). The different prototypes have been developed in parallel from the beginning of the process and they haven't been supposed to represent any final or complete stage (Bardram et al. 2004). They have been used as tools for the design and development of the PalCom open architecture. The details of experimental plan that has been carried out will be described in more details in Chapter 5.

The different methods reported above resulted in a number of different activities that will be more thoroughly addressed in the next two chapters. Among the most important there are:

- User workshops and activity modeling have been carried out in order to capture and evaluate a joint understanding of the work, the practices and the tools utilized, through discussions, focus groups, brainstorming sessions and various experiments.
- Different kinds of scenarios have been developed that interpret the user requirements that have emerged through fieldwork activities, by shaping the envisioned system in the daily practice of therapy in the water.
- Different data sessions have been conducted with the stakeholders, in which ethnographic data capable of revealing key aspects of user activities has been reviewed and analyzed.
- Different prototypes workshops have been performed to simulate design solutions by using low-fi and working technology prototypes.
- A Game Logics workshop has been carried out in order to design the distributed logic that characterizes the software development of the games.
- A 'Traveling Architects' workshop (Corry et al. 2006) has been conducted in order to conform software development to user requirements, as they were elicited through fieldwork activities. Indeed, 'traveling architects' represents a group of architects responsible for maintaining the assets of software architecture in a distributed development project in order to design, evaluate, and enforce software architecture in active collaboration with developers and ultimate users.

1.3.2 Process

The research approach adopted in this study integrates the methods described in the previous paragraph. The different activities are at the same time parallel and interwoven by means of several iterative cycles oriented to the software architecture design and development. In fact, from the beginning of the project, the objectives of

developing (1) an open architecture for palpable computing and (2) a conceptual framework to understand the characteristics of palpable technologies and their use, have been pursued in parallel and contemporarily linked processes.

The outcomes from the fieldwork and participatory sessions have continuously informed the process of architecture development and evaluation and, at the same time, the developments on the architectural side have guided the exploration of the use practices that were going to be supported by palpable technologies. Iterative cycles of mock-up and prototype experimentation also characterize the overall process. The PalCom software architecture has been experimented both in laboratory software testing and in embedded and simulated applications following the scenarios selected through the process. In fact the experiments were inspired by the development of the related activity, envisioning, prototype and quality scenarios.

The developmental process followed in the project is also incremental, in the sense that it went on improving the outcomes further and further. Since the processes have not been carried out in a sequential way, but rather throughout iterative cycles, the achievements have been continuously evaluated, improved and re-used to feed other processes.

The process described in this paragraph leads to the development of a range of prototypes of palpable applications and gains a firm understanding of a range of practices into which palpable technologies have been introduced and experimented. The process consists of the following main parts:

- The *fieldwork* analyzes current practices of achieving *palpability* among people and between people, their environments, tools and technologies.
- The *scenarios* provide narratives that describe how PalCom technologies may support, develop and enhance people's experiences and interactions in and with their environments.
- The *application prototypes* embody current PalCom design ideas and give a feeling for how they might be experienced and appropriated.
- The *architecture* of the application prototypes extends this experience of technologies in use to the architectural level.
- The *general architecture* provides a more generic viewing angle and sheds light on requirements and methods for a broad framework for making palpable systems.

The first iteration took place at the beginning of the project when different application sites were explored in order to evaluate how they might challenge and support the development of conceptual framework and the software architecture.

As the application scenarios have been explored and have evolved into early application prototypes, the initial formulation of the PalCom conceptual framework advanced in parallel. The research challenges, described in the introduction of the thesis, were adopted as a lens to observe the use practices and received meaning from the fieldwork. In this phase the notion of palpability - the core of the PalCom vision on Ubiquitous Computing - has been formulated in a preliminary way as a property of those tools that can be easily understood, managed, adapted and used in a purposeful way.

As already introduced in the previous paragraphs, PalCom addresses a set of key challenges in Ambient and Ubiquitous Computing. In this section we describe those challenges that we primarily deal with, by connecting use with technology (PalCom Deliverable 20). In particular, the Active Surfaces application prototype addressed the PalCom challenges in the initial phases of the process by providing a context to explore and evaluate valuable solutions for the development of small, resource constrained and embedded devices out of the architectural platform. Active Surfaces specifically addresses the following key challenges:

Scalability / Understandability. Highly scalable embedded devices challenge both the software architecture development (e.g. footprint sizes and functions) and the comprehension by the end users throughout their activities.

Construction / Deconstruction. The connections among distributed devices are allowed by the creation of dynamic, ad-hoc networks, which in principle may support the creative initiatives of the users. Software components and hardware devices are distinguished entities that fit into a common component model, which provides flexible interfaces and allows for dynamic composition and decomposition by the users.

Automation / User control. Ubiquitous systems require a certain degree of self-configuration due to the possibly high number of interconnected devices. This is probably beyond the reasonable limit of manual configuration, and the automated mechanisms also allow for split device solutions by which functionality comes from the composition of different entities, each contributing its share to achieve the common goal. It concerns the balance between system intelligence and automation and user control and preference.

Change / Stability. Stable and, at the same time, adaptable systems describe well the necessities of the rehabilitation practice. The therapists would deal with the configuration of the tiles according to the needs of the different patients while patients would be involved in featuring changes in the system. The system itself would guarantee a proper level of stability and persistence.

These are among the research challenges that have guided the fieldwork carried out at the application sites. They provided the dimensions through which to organize the data and the information coming from the field; the set of dichotomies provided a structure through which key attributes for the software architecture have been developed. A process of reinterpretation of the challenges took place and informed the future investigation of Use Practices and Architectural Qualities.

The iterative cycles and the dependencies among the processes have been represented in the figure below.

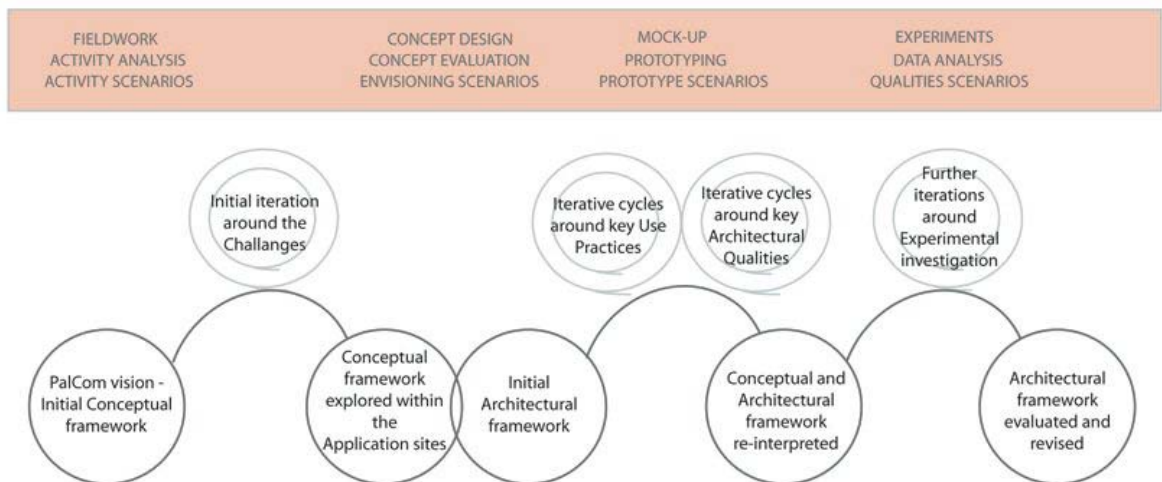


Figure 5. Iterative cycles through the research process and the activities carried out.

Throughout the second iterative cycle the conceptual framework already specified within the application scenarios has been further developed together with the initial formulation of the middleware architecture.

In the Activity Scenarios development the outcomes of the initial iterative cycles have converged and have been further elaborated around the challenges. In fact, the Activity Scenarios have been used to explore what the dichotomies might mean and how they can be interpreted in the proper context of use, allowing also the creation of new meanings to further expand the challenges themselves.

In the multiple iterative cycles of the process followed in this research, the Activity, the Envisioning, the Prototype and the Qualities scenarios have been used to bridge the use practices and the conceptual and architectural development. This is among the central research issues of PalCom. We try to move the vision of UbiComp from the notion of calm technology to engaging user experiences. This implies first understanding human activity. We used the Activity Scenarios to understand, as thoroughly as possible, what is relevant and appropriate in the specific domains of use, which in this case study was the therapeutic practice in the water. This work has been conducted together with the therapists of the Functional Rehabilitation Unit at the 'Le Scotte' Hospital, Siena, the water therapists from the D. Chiossone Institute, Genova and the trainers from the Disabled Children Parents' Association, Siena.

The development of the software architecture has then been continuously evaluated with respect to the key Qualities and Practices. Indeed, different architectural and application prototypes have been experimented by adopting the Envisioning, Prototype and Qualities scenarios as design and evaluation tools. The prototypes played a fundamental role as artefacts that mediated between the therapeutic practices and the architectural development. This part of the process will be explained in greater detailed in Chapter 4. To summarize in this process the design and development of the software architecture relies on the initial conceptual framework, how it has been revised throughout the application scenarios and how this process resulted in design solutions.

Chapter 1

Architectural Qualities and Use Practices

2.1 Play and therapy in the water

The Active Surfaces case study deals with the exploration and the design of innovative technologies in support of the practices of rehabilitation in water for disabled children.

In a domain in which the needs of the users, whether it be the therapists or the patients, are heterogeneous, and the context dynamically modifies itself, the introduction of Ubiquitous and Ambient computing technologies represents a challenge and an opportunity for the research and for the evolution of the rehabilitation practices. In particular, the introduction of UbiComp in the rehabilitative context presupposes and contributes to a specific reflection regarding the environment, the objectives and the instruments utilized in the rehabilitation interventions.

The interventions in water are, in fact, characterized by manifold factors, such as the psycho-dynamic characteristics of the water, the particular profiles of the recipients of the treatment, the needs and the abilities of the disabled children and the richness and the potentialities of the aids used in the practices. (Belloni 2007; Gamelli 2001).

The activities in water for the disabled are currently characterized by non-structured games in which the children can explore the aquatic environment and learn to move, float and swim. The activities in water are particularly well adapted to disabled children, since the water is a facilitating element; the ideal context for going beyond one's own limits, whether they be physical or psychological. In the water psychomotor therapists, teachers and educators work with children with motor disabilities and mental retardation, with the aim of intervening on the relational, sensorial and functional aspects and on the particular needs of the disabled child (Belloni 2007).

The therapeutic practice in water has as its objective the consolidation of the available cognitive and motor abilities and the stimulation of the global development of the child. This is achieved through functional activities that support the experiences of coordination in space and time, equilibrium, postural adjustment and dynamic/general coordination. (Belloni 2007).

Another very important aspect of the therapeutic interventions in water is the work that is done on muscular tone. This permits the child to balance their physical instability and also to influence the difficulties they experience with managing their attentive and cognitive capacities. With this therapy the water element is utilized to progressively permit the child to feel comfortable and to express their individuality. Thus in the water the dynamics of relation, equilibrium, movement in the space and in the dynamics of perception change (Lapierre 2001). Global interventions such as those that take place in the water then support the development of the child's central nervous system, favouring a balanced development of the mind and body.

This process can be supported by the use of instruments specifically designed for water therapy. Through the movement and the physical manipulation of objects, children with varying degrees of ability (multiple disabilities, physical-motor deficits and cognitive retardations) can progressively achieve a greater knowledge of their own bodies and of their own capacity for action.

In this chapter the conceptual basis of the therapeutic interventions in water (Par 2.1.1), the objectives of water activity (Par 2.1.2), the users and the disabilities (Par 2.1.3) and the existing approaches (2.1.4) are described.

2.1.1 Conceptual basis

Each human activity cannot be considered without first reflecting on the individual bodily skills. Given that a physical body is present, every interaction requires bodily skills. These skills are always a result of previous interactions with people, with the artifact in question and/or with other artifacts. These skills derive from having a physical body and they change the physical body itself as they develop. Beneath every specialized skill, there is always layer upon layer of bodily skills, which have been acquired throughout years of having a living body in the physical world.

The development of the self is described as a process throughout which the physical self, the social self, the teleological self, the intentional self, the representational self and the autobiographical self develop as layers, one upon the other, which are simultaneously composed of other layers (Gergely 2002).

The recognition of the self as a *physical agent*, is the first out of five different levels of the development of understanding agency and selfhood as they are distinguished in developmental psychology.

The faculty of acting, the state of being in action can be defined as agency (Dennet 1987). The development of a sense of physical agency involves the representation of both the causal relationships between the *physical self* (Gergely 2002) and actions, and actions and the external world. Infants learn very early on to differentiate between stimuli belonging to the self versus the environment. They are highly sensitive to the contingent relations between their motor responses and consequent stimulus events (Bahrick, Watson, 1985; Rochat, Morgan, 1995; Watson, 1972; 1994). According to this perspective, there is converging evidence that infants during their first six months of life come to represent their bodily self as a differentiated object in space, which can initiate action and can exert a causal influence on the environment (Gergely, Watson 1999).

The development of the physical self is thus characterized by the acquisition of self-detection, self-orientation and a proprioceptive recognition of physical agency. The representation of the self as a physical agent involves relations among agents, actions and outcomes on the other factor (i.e consequent changes in the environment).

Differentiating the representation of the body as a separate, integrated, and dynamic entity that can cause physical changes in the environment is an early challenge for children with special needs.

It is also the main area of intervention for therapeutic and rehabilitation treatments in the case of moderate or severe motor and cognitive impairments. The lack, or only the partial development, of a required skill makes certain operations inaccessible to us. In cases where we do not have all the means of knowing the structure of the world nearby, a lack of skill may restrict our view of the state-space in daily problem solving activities.

In understanding the theoretical basis underlining many rehabilitative approaches to disabilities, the concept of the Zone of Proximal Development (ZDP) is central, dealing with the potential skills *waiting* to mature in the individual (Vygoskij 1978). Learning and therapy are both seen as a voyage through the zone of proximal development. People may be understood not in terms of what they are but in terms of what they are becoming. This idea is particularly fruitful in the perspective of development of the self. Children may be supported and aided to develop *layers* of abilities in all the diverse forms and individual paths in which this may happen.

The concept of the zone of proximal development has been widely applied outside the areas of learning, developmental psychology and HCI (Engeström 1987). In all of these contexts, the zone of proximal development has come to mean the possible future practices, or developmental potential of the individual.

The therapeutic intervention is always a scaffold that enables the individual to reach their next ZPD. Therapists working with disabled children know how to provide this scaffolding (Wood et al. 1976) throughout the therapeutic sessions with the child. The processes of creating the scaffolding consists of support for development provided in tasks that may be novel and challenging.

Scaffolding captures the notion of reciprocal exchange, where participants adjust the way in which they understand something in relation to each other (Valsiner, 1984). Scaffolding is thus a process that changes. The content and the structure of the interactions between the person doing the scaffolding and the person being “scaffolded” changes over time in relation to the strengths and needs of the person.

The process of scaffolding that the therapist provides (Renninger, Granott 2005) acknowledges that (a) change is dynamic; it occurs over time, in forward and backward movements; (b) the nature of the tasks that are “scaffolded” vary, both in terms of their intentions and also how these are understood by participants; (c) children bring different characteristics to the particular experience of scaffolding, e.g., beliefs, interests, experiences, roles; and finally (d) the socio-cultural context in which scaffolding takes place may impact

- how the exchange and the task are understood,
- whether the scaffolding makes an impact,
- whether change is sustained even when the “scaffolder” is not present.

Limitations in individual functions may affect the ability to manage the stress that is caused by some particular situations. To ensure that the activity proposed to the child fosters the development of skills and social interactions, it is necessary to know what components may cause the child to feel stress, in order that they may be avoided. The precondition of an optimal therapeutic intervention is a balance between perceived challenges and perceived skills: the optimal stimulation (Hunt, 1965).

When perceived challenges and skills are well matched, attention is completely absorbed. This balance, however, is intrinsically fragile (Brooks, Petersson 2005).

For therapeutic treatments to be supportive in this sense, they must engage the child in challenging ways. Even though the activities provide children with a sense of challenge, they have to feel that their skills meet the challenges. When there is an imbalance between the challenges and the child’s skills, the child will become stressed or bored. If challenges begin to exceed skills, one typically becomes anxious; if skills begin to exceed challenges, one relaxes and then becomes bored. The therapeutic activities have to be intrinsically rewarding and the enjoyment derives from the game-playing style of the activities. The engagement in therapeutic playful activities provides the motivation needed to manage the physical and cognitive efforts, which are required to carry out demanding tasks involving stress, distraction, or crises, such as completing multiple tasks or being involved with social collaborative or competitive tasks.

2.1.1.1 Early field exploration

We directly observed motivational dynamics in the early field exploration at the Functional Rehabilitation Unit, Le Scotte Hospital, Siena. By mean of methods as observation, video recording, interviews, and activity analysis, we tried to understand the properties of the rehabilitation practice, the main tools adopted and the engagement of the patients. We especially dealt with the cognitive and physical rehabilitation held in the hospital gym and in the ward.

We were interested in understanding the therapists define individual therapeutic objectives for each patient (e.g. socialization, spatial orientation, personal autonomy) and how they try to accomplish these results along the duration of the treatment. Rehabilitation therapy takes place in sessions of one hour in predefined therapeutic treatment cycles. Following the objectives for each cycle, the therapists plan the sessions in

order to increase the difficulty level step by step, setting the patient appropriate tasks. Therapists at the hospital use different tools for psychomotor and occupational therapy such as wearable weights, bikes, pillows, etc. to improve physical-motor abilities and train movement and coordination. The development of these skills requires mid or long term practice and the tasks are measured in terms of speed, precision, distance, procedures, or techniques in execution.

They also use software applications for rehabilitation exercises based on tasks such as image matching, spatial orientation, and pattern recognition. To improve cognition, the therapists define specific goals such as spatial orientation (e.g. topological tasks), visual memory, procedural memory and social skills.

We outline two examples that clearly show the lack of integration between these two different kinds of therapy, cognitive and physical. Both of them are reported in PalCom Deliverable 33.

The first example refers to the educational software games used to achieve cognitive objectives. Among other activities with cards, pictures, photos and common objects, therapists also use the videogames to improve skills in image recognition, image matching, visual memory and learning procedures. The screenshots in the figures below show different kinds of videogame exercise, which engage the child while performing difficult memory and recognition tasks. The games below have different levels of increasing complexity all based on matching portraits pairs that have to be discovered.



Figure 6 Screenshots from portrait pairs based on pictures (8 cells)

The other example requires the patient to remember and find the letter pairs in order to uncover the image underneath. They interact by clicking on the cells with mouse or keyboard, and have the possibility to reflect upon their choices. They may play competitively in groups, trying to go faster than the others in accomplishing the task. The mosaic table is built of different number of cells (8, 18, 36 cells) that cover letters and image. Depending on its complexity, the task involves several skills such as visual memory and short-term memory.



Figure 7. Screenshots from games based on letters (36 cells)

On the other hand, the second example regards the field of motor physical treatment, the therapists mainly adopt system of rehabilitation exercises like the Bobath method. The Bobath method addresses the treatment of neurological disabilities. It has a theoretical basis in neurophysiology. Bobath exercises are

designed to inhibit spasticity and to aid in the development of new reflex responses and equilibrium reactions. The exercises aim at modifying postures progressing from simple to more complex movements in a sequence based on the neurological development of the infant.

The main aim of treatment is to encourage and increase the child's ability to move and function in as normal a way as possible. Here, it is fundamental to help the child to change his abnormal postures and movements so that he or she is able to develop better functional skills. The method consists of training the child to acquire key behavioural patterns of movement and positioning like head control, grasping, trunk turning, etc. In both the educational software and the Bobath method the motivational aspects are central. 4-5 year-old children are often very attracted to working with the PC, more than with common objects. Although software games really enhance motivation and engagement, the children are constrained to sit at the table in front of the screen. The activity exclusively involves their cognitive abilities.

When adopting the Bobath method in the physical rehabilitation treatment, the therapists usually utilize several objects to engage the patient in tasks that are very demanding for them. The treatment aims to support the development of motor and postural basic schemas through specific body movements facilitated by the therapists and dolls, coloured pillows or balls are used to encourage grasping and throwing, or sound objects to grab attention.

Differently from what happens in these kinds of treatment we would like to explore the richness of integrated physical and cognitive activities. Physical activation and physical agency are key developmental objectives to be pursued with personally meaningful and cognitively stimulating tasks. As long as patients improve their skills and become able to accomplish new tasks, therapists define goals of increasing complexity, e.g. combining different sensorial channels or merging cognitive and physical rehabilitation.

In the therapeutic practice there is great potential in using UbiComp technologies made palpable since the versatility it may allow, the different types of activities that can be carried out according to the type of planning, and the adaptation of the system's configuration to individual needs. Some experiences reported in the literature point out that such aspects related to therapeutic technology may help to increase the motivational drive of the subjects involved (Brooks 2002; 2006; Ellis 1995; 2004).

2.1.2 Main objectives

The main objective of the therapeutic intervention in the water that we primarily address in this thesis is to acquire the specific and global skills related to:

- Movement
- Manipulation
- Social coordination
- Physical dialogue

In many of the objectives described below, the intervention begins with the acquisition of basic skills, e.g. learning elementary and purposeful actions, in order to subsequently reach complex skills, such as integrated sets of actions, following rules and sequencing and coordinating actions. Each objective is specified according to issues related to different degrees of complexity (from very basic to complex skills).

Movement

- Walking: Moving along a surface on foot, step by step, so that one foot is always on the ground, such as when sauntering, walking forward, backward, or sideways.
- Moving around: Moving the whole body from one place to another; without changing body position, while sitting or lying, such as sliding along a bench, or by means different than walking, such as scampering, jumping and swimming.

- Moving around using equipment: Moving the whole body from place to place, on any surface or space, by using specific devices designed to facilitate moving or create other ways of moving around, such as with scuba equipment.

Manipulation

- Lifting up and putting down objects: Raising up an object or taking something from one place to another, such as lifting a cup and lowering it to the ground.
- Carrying: Taking or transporting an object from one place to another using the hands, the arms or the shoulders, hips and back, such as when carrying a drinking glass or a large object.
- Carrying, moving and pushing objects with lower extremities: Performing coordinated actions aimed at moving an object by using the legs and feet, such as kicking a ball or pushing floating objects.
- Hand and arm use: performing the coordinated actions required to move objects or to manipulate them by using hands and arms, such as reaching, turning or twisting an object.
- Pulling and pushing: Using fingers, hands and arms to bring and to move away an object, or to move it from place to place, such as pulling a string or closing a door.
- Throwing and catching: Using fingers, hands and arms to lift something and propel it with some force through the air, and to grasp a moving object in order to bring it to a stop and hold it, such as when tossing and catching a ball.

Coordination

- Carrying out (simple or complex) coordinated actions related to a single (or multiple) task(s), such as initiating a task, space and materials for a task, pacing task performance, completing and sustaining a task. And carrying out tasks in sequence or simultaneously.
- Carrying out tasks in coordination with other people, by playing collaborative or competitive games.
- Responding to criticism and social cues in relationships and using appropriate physical contact in relationships.
- Initiating and responding reciprocally to bodily contact with others, in a contextually and socially appropriate manner.

Physical dialogue

- Comprehending the literal and implied meanings of messages conveyed by gestures, behaviours and actions such as realizing that a child is tired when she rubs her eyes or that a warning bell means that there is a fire.
- Communicating with body language: comprehending the meaning conveyed by facial expressions, hand movements or signs, body postures, and other forms of body language.

2.1.3 Target users

This paragraph describes the target users identified for the treatment in the water according to the different impairments they may have. The descriptions below correspond to the International Classification of Functioning (ICF), launched in 2001 by the World Health Organisation, that defines human functioning as an indispensable relationship between the health conditions of a person and his or her ability to take action and participate. The ICF assumes that disability is a natural and common experience of living, not necessarily depending on illness.

In the present research, the ICF for children and youth (ICF-CY), an adaptation of the ICF for universal use in health, education and social sectors for children and youth has been utilized. The first release of the ICF-CY was launched only in October 2007.

It provides taxonomy for coding all health-related experiences based on an integrated biological, psychological and social approach (cfr. Varela et al., 1991). The same approach will emerge throughout the following chapters as a common strategy for laying out the experiments and for informing specific choices in the design process (e.g. the choice to design in support of certain abilities/ limitations instead of specific user groups).

The ICF has also influenced therapy programs in moving away from a ‘consequences of disease’ classification to one that acknowledges multiple factors contributing to a child’s health. This resulted in a shift of therapeutic focus from that of preventing disease to that of maximizing overall health (Kelly, Darrah 2005). Basing the new classification system on function rather than disability, in other words, starting with the functioning rather than the non-functioning, constitutes a real turnaround. This is of great significance for designing aids and activities for people with special needs: to start from a person’s abilities rather than from their inabilities in order for development to take place (Jönsson 2006).

The individual functions undertaken by this research are those that are specifically addressed and treated in the water therapy. They are grouped into global mental functions, specific mental functions, sensorial functions and body related functions. Not all the factors specified in the ICF-CY have been considered, only those that relate to physical agency and to the objectives of the research (i.e. design of ambient and ubiquitous systems) and those that make sense for the activities taking place in the swimming pool have been described. Some names have been adapted in order to agree with the rest of the language utilized in this manuscript: in these cases the original names are provided in brackets.

Global mental functions

a) Orientation functions

The Functions of knowing and ascertaining one's relationship to object, self, others, time and space. They include orientation (and disorientation) to time, space, place and person; orientation with self and others.

b) Behavioural functions (Intra-personal functions)

The different dispositions to act or react in a particular way, including a personal, behavioural style that is distinct from others.

- Adaptability: the disposition to act or react to new objects or experiences in an accepting manner rather than a resistant manner. Adaptability problems regard the acceptance of new objects and situations and of activities involving the willingness to act.

- Responsiveness: the disposition to react in a positive rather than negative manner to an actual or perceived demand.

- Activity: The disposition to act or react with energy and action rather than lethargy and inaction.

c) Motivation, energy and drive functions

The functions that produce the incentive to act; the conscious or unconscious driving forces for action. «Motivation involves self-generated curiosity and desire for mastery. This construct includes persistence, mastery for the sake of competence, preference for challenging tasks, goal- directed behaviour, short latency to task involvement, and positive affect while engaged in tasks» (Vig, 2007).

Specific cognitive functions

a) Psychomotor functions

- Psychomotor control: functions that regulate the speed of behaviour or response time that involves both motor and psychological components, such as in the disruption of control producing psychomotor retardation (moving and speaking slowly; decrease in gesturing and spontaneity) or psychomotor excitement (excessive behavioural and cognitive activity, usually non-productive and often in response to inner tension as in toe-tapping, hand-wringing, agitation, or restlessness).
- The quality of psychomotor functions: functions that produce nonverbal behaviour in the proper sequence and character of its subcomponents, such as hand and eye coordination, or gait.
- The organization of psychomotor functions: functions that produce complex, goal oriented sequences of movement.

b) Attention

- Maintaining attention: functions that produce concentration for the period of time required.
- Shifting attention: functions that permit the refocusing of concentration from one stimulus to another.
- Dividing attention: functions that permit the focus on two or more stimuli at the same time.
- Sharing attention.

c) High-level cognitive functions

The functions including complex, goal-oriented behaviours such as decision-making, abstract thinking, planning and carrying out plans.

- Organization and planning: the functions of coordinating parts into a whole, of systematizing; the mental function involved in developing a method of proceeding or acting.
- Cognitive flexibility: the functions of changing strategies, or shifting mental sets, especially as involved in problem-solving.
- Judgment: functions involved in discriminating between and evaluating different options.

d) Perceptual functions

The functions of recognizing and interpreting sensory stimuli: auditory, visual, olfactory, gustatory, tactile and visual-spatial perception.

- Auditory perception: the functions involved in discriminating sounds, tones, pitches and other acoustic stimuli.
- Visual perception: the functions involved in discriminating shape, size, colour and other ocular stimuli.
- Tactile perception: the functions involved in distinguishing differences in texture, such as rough or smooth stimuli, detected by touch.
- Visual-spatial perception: the function involved in distinguishing by sight the relative position of objects in the environment or in relation to oneself.
- Haptic perception: the functions involved in distinguishing shapes and forms, such as reading Braille.
- Proprioception: the functions involved in the awareness of limb positions and body boundaries.

e) Emotional functions

The functions related to the feeling and affective components of the processes of the mind, including the appropriateness of emotion, the recognition and regulation of emotions; the flattening of affection.

- Appropriateness of emotion: the functions that produce a feeling or affection congruent to the situation, such as happiness for being rewarded.
- Regulation of emotion: the functions that control the experience and display of affection.

- Range of emotion: the functions that produce the spectrum of the experience of arousal of affection or feelings such as love, hate, anxiousness, sorrow, joy, fear and anger.

f) Sensory functions

This paragraph is about the functions of the senses, seeing, hearing, tasting and so on, as well as the sensation of pain.

- Sight functions: the functions relating to sensing the presence of light and sensing the form, size, shape and colour of the visual stimuli. Including: visual acuity, visual field, quality of vision, sensing light and colour, visual acuity of distant and near vision and impairments such as myopia, hypermetropia, colour-blindness, tunnel vision, etc.
- Hearing functions: the functions related to sensing the presence of sounds and discriminating the location, pitch, loudness and quality of sounds. It includes auditory discrimination, localization of sound source, lateralization of sound, etc.
- Touch function: the functions of sensing surfaces and their texture or quality, including functions of touching, feeling of touch; impairments such as anesthesia, tingling, paraesthesia and hyperaesthesia.
- Sensitivity to vibration: the function of sensing shaking or oscillation

g) Body related functions (Neuro-musculoskeletal and movement related functions)

The functions of movement and mobility, including the functions of joints, bones, reflexes and muscles.

- The functions of the joints and bones
- Muscle functions
- Movement functions

At this global level it is important, however, to emphasize that children with motor disabilities often can develop skeletal and muscular complications such as inadequate stretching, constant muscular contractions, or inflammatory disorders of the joints, soft tissue or skin (Hornof & Cavender, 2005).

2.1.4 Two main approaches

Children usually see the swimming pool as a fun environment: being in the water decreases the effects of gravity, thus making the activities involving balance and coordination more attainable for the child. In the water the child with special needs can increase their range of motion and coordination but also improve independence and sensory processing. The water is an excellent medium for addressing the physical, cognitive, and psycho-social needs of children (Belloni 2007). It can offer the freedom of movement without the restrictions put into place by gravity, as well as opportunities for increased self confidence, risk taking, learning, socializing and having fun while addressing therapy goals.

Children with different diagnoses can benefit from aquatic therapy: Cerebral Palsy, Down 's syndrome, Autism, Neurological Disorders, Perceptual difficulties etc.

Indeed water activities provide proprioceptive and tactile input. This is very important for children who have significant sensory difficulties, and are very distractible. These children over or under react to stimuli and have very strong reactions to certain textures. The water provides a safe and supported environment, which not only supports them, but also provides them with hydrostatic pressure that surrounds their body in the water. This pressure actually soothes and calms the children, providing the necessary sensory input they crave.

There are currently two main approaches that address the therapeutic practice of treating disabled children in the swimming pool: one is the Psychomotor Therapy, a relational approach characterized by exploration, symbolic play and play activities, and the Aquatic Therapy, which is mainly characterized by structured programs of exercise and instructions. A comparison between the two approaches is made below.

Psychomotor therapy	Aquatic therapy
Non-specific intervention	Specific intervention
Unstructured activity	Structured activity
Play	Rules
Empathy	Efficiency
Symbolic use of materials	Functional use of tools
Exploration	Exercise plan
Integration	Body schemes
Socialization	Individuality
Mediation	Measurement and assessment scale
Engagement	

A survey on the most common therapies in the water has been carried out. In the following sections the therapies pertaining to the two approaches have been described.

2.1.4.1 Psychomotor therapy in water

The psychomotor approach was born in opposition to the traditional concept of physical education as training of the body finalized for the improvement of the executive factors of movement, control and motor productivity. The psychological dimension of the movement, as well as the affective, cognitive and relational components, are privileged (Le Boulch 1975).

Psychomotorism was established in France as an autonomous discipline in the 1970's, thanks to its founders B. Aucoutourier, A. Lapierre, P. Vayer and J. Le Boulch.

It draws from the theoretical approaches of various scientific-cultural disciplines: behaviorism, neuro-physiology, psychology and psychoanalysis, cognitivism (Piaget 1962), characterology (Wallon 1959), philosophy and etiology. Considering the psychomotor life of the children, as it is connected to affectivity and to the body in its globality, a single method of psychomotor treatment or technique does not exist in psychomotorism.

According to the school, variations are favoured that diversify the various methods:

- *Psychomotor re-education*: this is imposed in the cases in which the instrumental deficit predominates, with the risk of secondarily starting relational problems. The relational approach of A. Lapierre (2001) and the Body Schemes intelligent learning of P. Vayer (1971) belong to this school.
- *Psychomotor Therapy*: regards in particular all of the cases and problems in which the affective or relational dimension would seem dominant at the onset of the disturbance. It can be associated with psychomotor education or continue with it. The psychomotor practice of B. Aucoutourier (Aucoutourier et al. 1984) belongs to this school.
- *Psychomotor education*: a basic training for all of the children (both normal and in handicap situations), this approach responds to a double necessity in as much as it assures the functional development while taking the child's possibilities into account, and helps their affectivity to

manifest and balance itself by means of the exchange taking place with their own human environment. The functional approach of J. LeBoulch (1975) belongs to this school.

Relational therapy (A. Lapierre)

It poses a decisive importance on the child's active participation in the therapeutic action. The context must provide useful feedback, but also ensure that the child is stimulated to enter the situation and become a leading figure, not the user of a service. This is obtained by re-establishing meaning in the activity: for the child, the best way to do this can only be play (Belloni 2007).

In the relational approach it is fundamental to engage children in play, either alone or with others, taking part in formal and informal play activities with toys and games. Children are occupied with objects, materials or toys in social play activities.

Psychomotor practice (B. Aucouturier)

The psychomotor practice affronts the treatment of disturbances in the affective and relational dimension. Through tonic-emotional dialogue and non-verbal communication, the educator acts as the original relationship between the mother and child. The therapeutic relationship between the educator and the child is therefore important, but not comprehensive. The method of the psychomotor practice is based on listening, non-directiveness and on the non-verbal communication. It aims at improving the interpersonal interaction by providing interaction with people in a contextually and socially appropriate manner, regulating emotions and impulses, controlling verbal and physical aggression, acting independently in social interactions, and acting in accordance with social rules and conventions.

Body schemes intelligent learning (P. Vayer)

Vayer's approach deals specifically with motor-functional deficits with the risk of relational repercussions, such as synkinesis, hypotonicity, dysphemia, tics and disgraphies. Techniques are utilized that are appropriate to the restructualization of the corporeal scheme (relaxation, development of space-time, laterality). In this method in particular the affective-relational dimension is considered secondarily. According to Vayer (1992), if the activity «is always the expression of an intention [...], the exercise will be useful only if the subject personally gets involved in the action, in the relationship; this implies that it has a meaning. Can it be imposed? One's desire can always be imposed on another, but we shouldn't be surprised if he/she reacts and develops defence systems. These reduce desire, therefore involvement in the activity and, as a consequence, the benefits that could be obtained».

Functional approach (LeBoulch)

The functional approach considers the person in its globality. It has as its objective the basic training that is indispensable for every child, whether they be normal or differently able. It proposes the assurance of the functional development, taking the individual potentialities into account, and helps their affectivity to expand and balance itself by way of relationship with the human environment.

By way of the practice of bodily work, it is possible to involve various functional systems such as the muscular and the nutritional systems (on which the "factors of execution" depend), which influence the motor productively, as well as the central nervous system, which coordinates the whole of the other systems and is at the same time the support for mental functions (cognitive and energetic-affective).

2.1.4.2 Aquatic therapy

The Aquatic therapy approach is mainly inspired by the idea of applying physiotherapy in the swimming pool by relying on the strengths of the water. Many of the treatment methods used justify a simplified interpretation of body movement, and the motor action is broken down into reflexes, motor units, and muscle bundles or, in the best case scenario, muscle groups. In this perspective a broad range of aquatic

therapy techniques exist to treat a wide variety of diseases.

Ai Chi (Salzman 1999) is a form of active aquatic therapy for people suffering from pain, such as back pain or headaches, stress-related disorders, such as anxiety, balance deficits, such as head injury and muscular-skeletal stiffness disorders, such as rheumatoid arthritis.

During treatment, therapists usually stand on a pool deck, and verbally and visually instruct you to perform a slow, rhythmic combination of therapeutic movements and deep breathing.

Aquatic Proprioceptive Neuromuscular Facilitation (PNF) (Salzman 1999) is a form of active aquatic therapy for people with neuromuscular deficiencies resulting from faulty development, trauma, or nervous and muscular-skeletal system disorders. During treatment, therapists verbally and visually instruct you to perform a series of functional, spiral and diagonal movements while you stand, sit, kneel or lie in water. You can perform the patterns actively, or with assistance or resistance from specialized aquatic equipment.

Bad Ragaz Ring Method (Salzman 1999) is a form of active or passive aquatic therapy that clinicians use on patients with bad spinal stabilization skills, neurological disorders, and lower extremity weight-bearing restrictions. It's also used on patients who can't immerse themselves underwater because of respiratory difficulties.

Therapists verbally and visually instruct you to perform a series of movement or relaxation patterns while supporting you with floats. You can perform the patterns passively (for flexibility and relaxation), actively or with assistance or resistance from providers.

Fluid Moves - Aquatic Feldenkrais (Salzman 1999) is an active or passive aquatic therapy intended for people with balance dysfunction problems, chronic pain or a restricted range of motion.

It's modeled after the Feldenkrais Method, which aims to make patients aware of their own body. They do this through a sequence of movements. During therapy, you follow movements based on the early developmental stages of infants. For instance, you might stand in chest-deep water, typically with your back to the pool wall, and perform a slow, rhythmic combination of therapeutic movements and deep breathing.

Halliwick Method (Salzman 1999) is a form of adapted aquatics for patients with orthopedic, neurological or development disorders, including arthritis and cerebral palsy. Therapists usually hold or cradle you in the water while they progressively teach you balance and postural control. This is done through a series of activities and games, which require more sophisticated rotational control movements. You're continuously required to react to, and eventually predict the demands of an unstable environment.

Watsu (Salzman 1999) is a form of passive aquatic therapy for patients who have a stress disorders or depression, restricted range of motion in their spine and other extremities, and high tone or spasticity problems. It's modeled after the principles of Zen Shiatsu (massage), which is the practice of pressing body points to direct the flow of chi, or energy, throughout the body. Based on these beliefs, Watsu therapists usually hold or cradle patients in the water, while moving one segment of their body, which results in stretching their other body segments and unlocking chi.

Task-type training approach (TTTA) (Salzman 1999) is a task-oriented approach that emphasizes functional skills performed in functional positions, in which patients are encouraged to be active participants in their skill development. TTTA is a set of principles that guides clinicians in designing treatment programs. Task Type Training Approach was first described as a way to teach functional activities to patients who had sustained a stroke. The principles can be extended to include the treatment of all patient disorders, particularly those involving neurological dysfunction.

In the table below, the outcomes of the survey have been summarized and the therapies have been organized in relation to the disabilities they aim to treat.

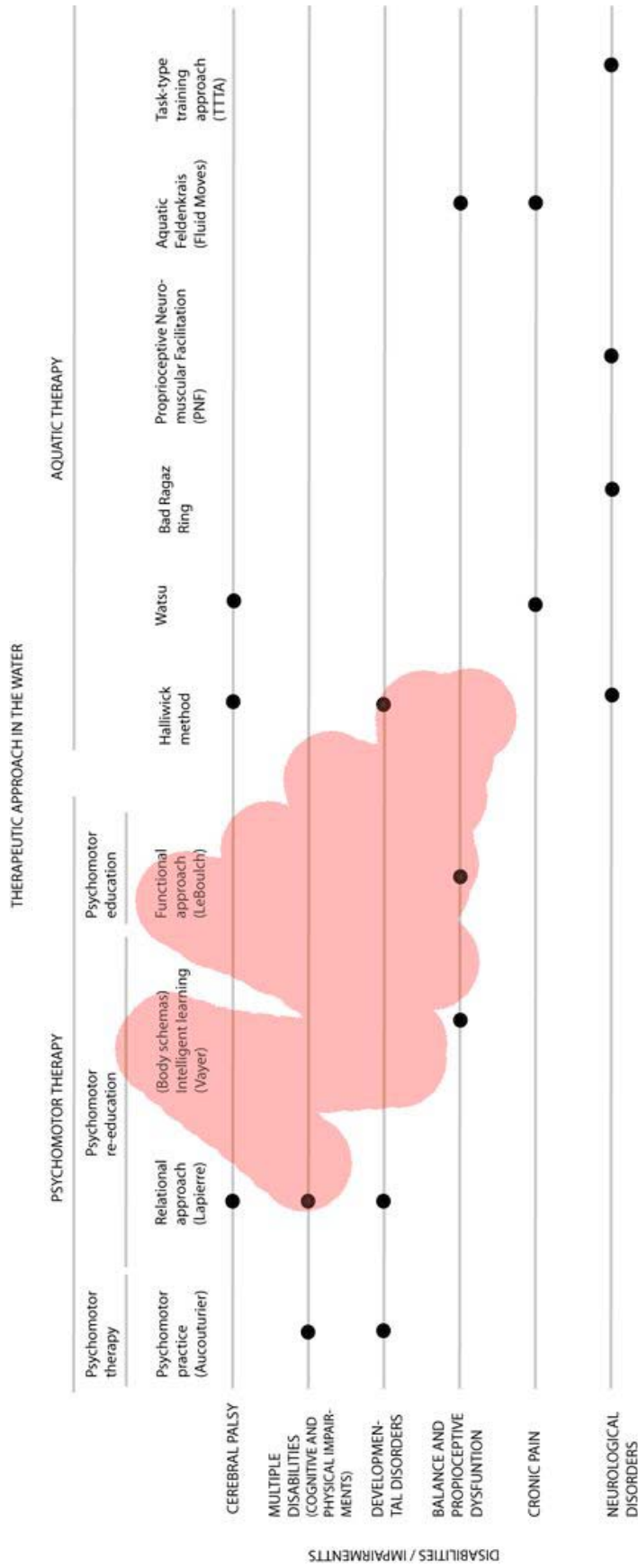


Table 8. The therapies and the disabilities they aim to treat.

The table is also useful for identifying the approach that we want to follow in this research and which grounds the design process. Both the two approaches related to the psychomotor education, guided by LeBoulch and Vayer, which are covered in the table with a red shape, inspired the design process. The shape also represents the opportunities we have to address open challenges and unexplored objectives with regard to specific target users (i.e. multiple disabilities, developmental disorders and balance and proprioceptive disfunctions). Thus, it identifies both the approach and the possible area of intervention for the design activity. This process is described in detail in the next chapter.

2.2 Enabling technologies

We address the domain of play and therapy in the water with disabled children from the specific perspective of interaction design, i.e. the focus on the use of tools, the adoption of technologies and on the practices with end users. It is thus necessary to survey the enabling technologies that have been used on children with special needs in therapeutic play.

More specifically, we take into consideration the technologies that enable use practices such as those described in Chapter 1, or the attributes that can be related to those. Notwithstanding the brief description of the systems and the applications, each project will not be treated as interesting from a technological point of view. We highlight rather the way in which they suit the practices of use that are addressed in this research.

For the sake of convenience, I have restricted the field by considering the interactive embedded technologies that are specifically designed for the therapeutic setting.

The field referred to as Therapeutic Technologies is not yet strictly defined and institutionalized. One attempt has been made at the MIT Media Lab Europe (<http://medialabeurope.org/research/other.php>), where a research initiative held this label until the institute closed. Even if only preliminarily explored, the therapeutic technology applications developed by the MIT initiative dealt with virtual reality and video games applications (Sharry et al. 2003). We don't report on these research studies, as we are mostly interested in how the embedded technologies may support the cognitive and the physical rehabilitation activities rather than the effect of the mere immersion in a virtual reality environment. Other criteria have been followed in this survey.

The research projects regarding Assistive technologies are also not considered because the goal of the present research is of quite a different nature in comparison to what is stated by the ISO-9999. In fact the aim of Assistive technology is "to prevent, compensate, relieve or neutralise the deficiency, the inability or the handicap". Finally we don't take projects regarding Educational and rehabilitative software, Multi sensory environments and Rehabilitation robotics into account, as a comparison between different paradigms lies outside of the scope of this research.

The research projects described in this survey are basically those ones that, in National and European research, primarily dealt with disabilities and special needs, with a specific focus on play and therapy. It must be noted that none of the currently existing research projects deals with designing therapeutic technology for the aquatic environment.

Two main applications, the SoundBeam and the Soundscapes technologies described below, evolved as a result of three European research projects. In the first, titled 'CARESS' (Ellis 2004) sonic elements (music scale tones) were used as feedback stimuli in response to sensors attached to the body and/or movement within a linear ultrasound sensor (Soundbeam). The second research project was Twi-aysi (The World Is as

You See It) (Brooks et al. 2002) within the European Network for Intelligent Information Interfaces (I3). This study was informed by the ongoing research titled SoundScapes (Brooks 1999, 2002, 2004a, 2004b, 2006), in which 3D non-invasive sensor technologies were developed to empower adults and children with severe disabilities to control audiovisual feedback by means of intuitive and natural gestures. The third project was called CAREHERE (Creating Aesthetically Resonant Environments for the Handicapped, Elderly and REhabilitation) under the 5th Framework IST (Brooks, Hasselblad 2004). CAREHERE deals with applications relating to persons with special needs through multimode stimuli feedback.

Soundbeam (<http://www.soundbeam.co.uk/>) is a musical instrument, similar to an invisible, elastic keyboard in space, which allows sound and music to be created without requiring physical contact with any equipment. The system utilizes ultrasonic ranging technology coupled with a processor that converts distance and movement information into MIDI. It has proven to be significant in the fields of disability and special education (Ellis 1995; Ellis 2004). Individuals who may be especially difficult to stimulate can benefit from what may be a novel experience of initiation and control. Soundbeam has been applied in 'sound therapy' with the goal of training and supporting posture, balance, and cause-and-effect, as well as in creative and experimental explorations in which disabled children and adults become the composers and performers.



Figure 9. Disabled child playing music by mean of the Soundbeam. (Source: <http://www.soundbeam.co.uk>)

Inspired by the Vygotskian approach (i.e. the Zone of Proximal Development), this project aims to enhance the individual capacities for action by Sound Therapy. This is enabled by the experience of control and awareness that Soundbeam supports (Ellis, Van Leeuwen 2000).

The Soundscapes (<http://www.soundscapes.dk/>) technology refers to a set of tools, assembled according to the user's objectives in a game-oriented, engagement environment (Brooks 2006; Petersson 2006). This allows opportunities for personalization and ad hoc adaptation. Created as a whole, the equipment is capable of capturing body function data information (for example, movement data, breath pressure and biofeedback) that is then used as a source for real-time interactive manipulation and control with multimedia. The capturing of the body function depends then on a variety of equipments: sensors, cameras, biofeedback and infrared beams.

The sensitivity of the areas can be adjusted and even small body part movements, such as a finger twitch, or full body movement, such as a standing balance rotation exercise, can be sourced for processing and effect the selected multimedia feedback, mostly video images (e.g. a Space shuttle) projected on the wall (Brooks et al. 2002). This supports a meaningful, non-verbal body language that enables moments of communication (Hasselblad et al. 2007). The different configurations enable the user/ therapist/ family member to select images and provide real-time manipulation. The system provides the opportunity to control and manipulate an external object without the requirement of any physical handling skills. This aims at providing an optimal scenario (cfr. 2.1.1) in which the challenges are perfectly balanced with the

competencies of the child.

In contrast to the two previous projects, the PLAYWARE project (<http://www.adaptronics.dk/Projects/index.html>) is an interactive playground made up of tangible tiles, which functions as a building block by containing processing power, sensors, actuators, and communication capabilities. It is an Ambient Intelligence application for play and therapy that can be adapted to user needs and supports the creative invention of games. The tangible tiles measure 25cm*25cm and have a force-sensitive resistor that registers when someone jumps on the tile. There are 18 red and blue LEDs distributed equally on each tile. The software runs on a micro-controller (ATmega128), registering the activity from the sensor and controlling all the LEDs individually. It communicates with the neighboring tiles and controls the games by way of a distributed processing; communication can be either wired or wireless. PLAYWARE can be used with different physical configurations on the walls and floor. Games for specific therapeutic treatments are downloaded into a master tile, which detects the tiles' structure and initiates the game. The tiles analyze patients' movements, measuring their progress. PLAYWARE is mainly designed for physical activity and learning through play (Lund, Jessen 2005). Built upon the concept of building blocks, the coordination of primitive behaviors through a physical module such as a tile (being a primitive behaviour), together with the interaction with the environment, decides the overall behaviour of the system. As patients step on or press the tiles with their hands, the tiles give feedback, indicating whether the pressure is firm enough, or if the user is moving quickly enough. Individuals can use the game alone, or up to four patients can compete against each other in a game.



Figure 10. Children playing with the Playware

The Music Toys is an application developed within the music performance & education Toy Symphony project (<http://www.media.mit.edu/hyperins/ToySymphony/musictoysmain.html>). The Music Toys is a series of special instruments that require no special skill, but which reward curiosity, imagination, and expression. The Music Toys are 'hyper'-instruments that can be intuitively learnt (Machover 2004). The Music Toys have been used as a comprehensive set of musical, neurological, and behavioural tools and techniques to investigate whether (and, if so, under what conditions) musical activities are associated with enhancements and improvements in memory, concentration, pain management, anxiety, stress, and creative imagination (Machover 2004). Music Toys technology aims at enabling people, at any level of physical or mental ability or disability, to express themselves musically. Music Toys new interfaces contribute to the creation of contexts through which the subject can participate actively, which can be activated by touch, gesture, whistle and hum. For example, patients with cerebral palsy may be able to control the system software (i.e. the Hyperscore) through an infrared sensor, a receiver running a mouse control program, and a serial connection to the computer. Also, patients with blindness may be able to communicate their

musical ideas physically, through tapping or vocalization,, since a series of musical relationships can be managed within the Hyperscore program. The work with the Music Toys confirms the importance of custom-tailored and highly adaptive technologies.



Figure 11. A quadriplegic patient playing music with Hyperscore.

Another therapeutic technology is the Rolling Pins (RP), a modular device designed within the Multi-sensory Room project (Marti et al. 2007) that is aimed at developing non pharmacological therapeutic protocols and IT (Information Technology) solutions for the treatment of dementia in institutionalized contexts. The Rolling Pin addresses the specific therapeutic area of the stimulation of social exchanges and the prevention of cognitive and behavioural decline.

The Rolling Pins are semi transparent plastic tubes capable of measuring their orientation and the speed of their rotation. At a local level they have three types of feedback: RGB light, sound and vibration. They can also be used to control ambient output: light and sounds. The peculiarity of the RPs is that they are able to communicate with each other or with other devices equipped with the same radio communication technology. The Rolling Pins can be configured in different ways with respect to user needs and therapeutic objectives allowing a broad range of possible configurations (input/ output modalities, interaction strategies). The opportunity to continuously adapt the difficulty of the task to the skill of the patient is fundamental to the creation of an optimal experience and to the maintenance the patient's attention. The Rolling Pins are now currently under experimentation with autistic children in therapeutic play settings.



Figure 12. Testing the Rolling Pins with elderly people.

In the following table the projects have been summarized according to issues like Therapy, Inspiration, Goals, Users and Use properties.

	THERAPY	INSPIRATION	OBJECTIVES	USERS	USE PROPERTIES	
ENABLING TECHNOLOGIES	SOUNDBEAM Caress project	Sound therapy	Vibroacoustic therapy	Expressive movement Coordination Control Communication	Learning disabilities Multipledisabilities Autistic disorders Neurological disorders	
	SOUNDSCAPES Twi-aysi project CareHere project	Play therapy		Expressive movement Control	Multiple disabilities	Configurability
	PLAYWARE	Physical play therapy	Physiotherapy	Physical activation Balance Coordination		Configurability Re-use
	MUSIC TOYS Toy Symphony Project	Music therapy	Music therapy	Expressive composition Communication	Learning disabilities Cerebral palsy	
	ROLLING PINS Multisensory Room Project		Relational therapy	Interaction Expressivity	Dementia Autistic spectrum	Configurability

Table 13. The Enabling technologies

The Use properties appointed in the table refer to those described at the beginning of the paragraph. As discussed above, the term “Configurability” has been used in this context, meaning specifically the propensity of a system for being configured ad-hoc and personalized according to user needs, to allow the construction of several devices in unique configuration. Independently of whether the described technologies may marginally support some of the key properties of Palpable computing, the table shows how none of the current projects on therapy and play specifically deals with key (architectural) Qualities and (use) Practices.

2.3 Fieldwork and Activity Analysis

Together with the study of the domain (Par 2.1) and a survey of the enabling technologies (Par 2.2), fieldwork has been carried out with the aim of directly exploring the field of therapeutic intervention in water. The fieldwork has been conducted in two settings for psychomotor therapy in water, the Disabled Children Parents Association, Siena and the D. Chiossone Institute in Genova. In the former context, trainers specifically trained for working with children with special needs mainly conduct the activities in the water. The activities are not structured and, while personalized to each patient, are not formally assessed. The latter is a rehabilitation center where the therapy in the water has been carried out in a more formal way by therapists specifically trained for treating cognitive and physical disabilities in the water. Therapists and trainers in both the settings followed a mixed functional-relational approach mainly based on playful activities and direct contact with the children.

The fieldwork provided us with a meaningful collaboration with trainers and therapists; with their valuable suggestions and comments they gave us a sense of the activities taking place and the challenges they pose. The participation of the actors resulted in a number of activities both during the fieldwork and the experimental phases of the process. Thus a variety of methods and techniques have been adopted, as described in Par. 1.3. We adopted ethnographic methods - such as field observation and interviews - and design methods - such as user workshops and creative brainstorming.

The ethnographic activities attempted to observe and reveal relevant issues related to the environment (the features of the water, the physical structure of the swimming pool), the actors (therapists, disabled children, parents), the tools (objects, toys and water noodles) and, above all, the activities (the procedures, the different phases, the practices). The outcomes of this activity resulted in a number of key observations that informed the whole design process. The development of activity scenarios summarized many of the outcomes described in the following sections.

Together with the research activity undertaken, I have been also directly experiencing the setting as an assistant to the activities in the water with children with special needs in the Disabled Children Parents Association, Siena. This experience provides me with the unique opportunity of addressing the challenges of working with disabled children, being responsible for the activity itself and experiencing how this job may be both rich and valuable, but also critical and demanding.

In the following sections the descriptions of the tools, actors and environment, the procedures and activity models and the summary discussion refer primarily to the research activity conducted at the D. Chiossone Institute, where the therapy in the water is part of a clinical rehabilitative plan, the activities are structured within a multifaceted rehabilitation plan and specific and coordinated objectives are pursued.

2.3.1 Context, Actors and Tools

Context

The context in which the activity takes place is essentially an immersive physical and emotional environment that is experienced with the whole body. The setting is the public or private swimming pools where non-specific or dedicated pools are used as therapeutic settings for people with special needs.

Special pools dedicated to disabled people generally differ from the others by the higher water temperature (around 30- 32°C) and the presence of equipment to aid disabled people with entering and exiting the water. These pools may also be of different dimensions and depth and may be characterized with respect to

the activity taking place within them. There may be lanes in order to favour the swimming or the walking of the disabled or all water surfaces may be available for social and symbolic games.

The water temperature is one characteristic that makes a noticeable difference in the treatment. The appropriate temperature has valuable effects on muscular tone and prepares to sustain demanding tasks and physical efforts. In fact, as described at the beginning of this chapter, entering the water means living the holistic experience of perceiving the whole body under different physical laws (e.g. buoyancy). This provides new and different capacities for action, especially for those who have severe physical and motor impairments. These people can particularly enjoy the perception of an uncommon autonomy and ability to move. Being in the water also alters the perspective on the surrounding environment. Depending on the position (if you lie or if you stand), you may change between different point of views on objects and persons.

The water environment is also immersive for what regards the lights - because of the reflections and transparencies - and the sounds - because of the echoes and the amplification of voices and noises due also to the presence of other people (parents, trainers, and people in other pools).

It is very difficult to communicate exclusively by voice, and that's why the interaction is also a more physical dialogue, based more on facial expressions, gestures and physical facts rather than words. In the water you may also feel the waves created by movement. The sensorial experience is thus predominant; sight, hearing, touch and whole body experiences are the key senses the therapists' play on throughout the therapy.



Fig. 14. The Acquacalda swimming pool in Siena

Actors

The main actors of this therapeutic setting are the children with special needs. Other actors involved are the therapists and caregivers, who take part in the activities, and the parents, who are usually present at the intervention, accompanying their children to the pool and helping them to enter and exit the water.

In this paragraph the needs of the main actors are described in order to provide insight on the nature of the disabilities treated and on the current therapeutic activities being used. Such information can hardly be expected from children with mental delays or cognitive impairments. The profiles briefly described below stem from interviews with caregivers and parents and from the available literature. These descriptions regard the real target users who may have the impairments and delays specified in Par. 2.1.3.

Children with very diverse profiles actually benefit from therapeutic play in the water. The users we have observed can be summarized in three main groups: children with cognitive and socio-relational disabilities, such as autism, children with physical impairments like cerebral palsy and those with a mental retardation such as Down syndrome.

Autistic Spectrum Disorders and Other Affective and Socio-Relational Disturbances

In general terms, the main disabilities that are characteristic of people with autism are impaired social interaction and social communication and having a limited range of imaginative activities. People with autism also typically show little reciprocal use of eye-contact, and have a tendency toward repetitive behaviour patterns and resistance to any change in routine. Some children with autism do not interact with other children on their own initiative, and although some of them can play interactive games with others if they are told to, they will need to be instructed and supported during the game, otherwise they very quickly return to their own solitary 'obsessive activities'. Children with other affective and socio-relational disturbances will also have these kinds of impairments, even if in different degrees.

Physical and Motor Disabilities and Cerebral Palsy

These children have limitation or an impossibility of movement, restrictions in force, abnormal postures, the presence of neurological movement disorders such as dystonia, tremor, ataxia, etc. Children with mild or severe motor impairments particularly benefit from treatment in the water. Children with CP can be severely impaired in playing by their motor disability, but also by speech and communication disabilities, and sensory impairments (visual and/or hearing). These children can differ a lot from each other because the degrees to which they experience disabilities vary, resulting in a large number of different levels of functioning.

Mental Retardation/ Intellectual Disabilities/ Learning Disabilities

Children with mental retardation also referred to as intellectual disabilities or learning disabilities (for example children with Down's syndrome), have a reduced capacity for attention and might not understand the meaning of the proposed activity. They might not understand the meaning of language and many of them have speech limitations too.

It is also important to focus on age ranges to be able to make sense of the practice. Current therapeutic play activities are mainly for children in the age range of 3-6 years, who would like to do things with others and experiment with movements and actions; and children in the age range of 7-13 years, who would also like to perform activities on their own, such as trying, inventing, building, even when nobody else is available to assist them. Note that the chronological age is not suitable to be applied to these target users without an understanding of their developmental level.

The therapists and trainers are the other main actors of this setting. They essentially have the role of facilitating the playful physical, social and emotional experience. They have to mediate the social relationships, the experience in the water and offer a reassuring presence to the child. They are the scaffolds that allow the child to express and freely explore the space of the pool. The therapists have to facilitate the activity, and not impose rules or, on the opposite extreme, abandon the child without a guide. Even when the child would like to explore by herself the therapist should also be present and support her independent action.

The intervention is considered successful when the therapist interprets the meanings of the behaviors of the child. Having an intimate knowledge of the child is central to achieving this interpretation.

Tools

The therapists use two different kinds of tools during the current practice in the swimming pool. There are not therapy specific tools and the professionals have to define their own set of tools to rely upon. Being responsible for the care of the child, they pay attention to the materials, to how they react to the water and the potential dangers they may bring. After this careful analysis they choose the common noodles and tubes for swimming pool and water resistant toys that can be particularly pleasing for children.

The noodles are usually used as aids for buoyancy and as a means for engaging in symbolic play, e.g. the tube frequently becomes the horse that brings the child to the end of the lane.



Figure 15. Tubes, noodles and other materials for the activity in the water.

The other kinds of tools are generally common toys that are water resistant and can be used in the water. They are special materials and toys available in the market and not specifically designed for the water environment or for therapeutic activity.

Those toys are generally made of coloured plastics and preferably have dynamic parts. An example is the polyp that floats and adapts to the surface of the water. The presence of these kinds of objects is also emphasized by the use of a torch for discovery games based on light.



Figure 16. The polyp

The most popular objects are those that produce some kind of effect and make use the dynamic property of the water. They capture the interest of the children and have been used for symbolic play or for play that involves physical causality. Those objects are also used for many of the game activities described in the next paragraph (e.g. give and take, push and pull, reaching with the extremities). Examples of these toys are the mill and the boats in the figures below.



Figure 17. The mill and the boats

It must then be noted that no interactive or digital tools have been used in the current practices examined.

2.3.2 Activity Modelling

The children enter the therapeutic treatment in the water by way of a coordinated process characterized by: The initial assessment focused on the profile of the patient (1), their history of previous therapeutic treatments, the observation of the mother-child relationship and the assessment through profile specific Scales, e.g. the Learning Accomplishment Profile (L.A.P), for learning disabilities, or the Renè-Zilkin Scale, for evaluated sight impairments.

The design of the therapeutic plan (2). The primary indication is to start from the needs of the child, which makes the identification of the rehabilitative project and, later on, the single rehabilitation sessions with the child possible. The rehabilitative project is defined by the group of therapists that usually treat the child (speech therapist, psychologist, psychomotor therapist and physician) and it consists of coordinated activities in different areas such as the visual stimulation, basal sensorial stimulation, physiotherapy, pet therapy and, among the others, the therapy in the water. In all of the treatments the objectives are defined by the relationship established between the adult and the disabled child within a rehabilitative context that makes use of play activities.

Throughout the treatment periodic assessments take place (3). They are based on protocols defined ad hoc with respect to the therapeutic plan. In fact specific protocols for the therapy in the water are not currently used in the practices we have observed. The therapy in the water is mainly assessed by the therapist's individual qualitative assessment, which is based on the global wellness of the child and how the acquired behaviours can be transferred to other settings (i.e. self confidence and self awareness). In this respect, the water context makes exception with respect to other common therapeutic settings. In particular, it is hard to evaluate whether some changes or improvements are due to the role of the specific therapy in the water, to the global treatment of the child (physiotherapy, pet therapy, etc.) or to the growth of the child.

To my knowledge there are not, in fact, scales to measure and assess the activities in the water. The qualitative criterion adopted is the overall improvement of the child. In particular, as other research suggests (Ellis, Van Leeuwen 2000), the therapist might observe if the behavior changes:

- from involuntary to voluntary
- from accidental to intended
- from indifference to interest
- from random to purposeful
- from exploratory to preconceived
- from isolated to integrated

These headings may summarize the overall picture of development for each individual (Ellis, Van Leeuwen 2000). Other criteria for assessing the outcomes of the intervention in the water are the improvements in the social interactions with peers, family members and operators and the increase in vocalizations or in the physical dialogue through the contact with the caregiver.

One of the outcomes of the fieldwork and activity analysis is the activity model, in which the different phases of the practice have been described in detail. We have addressed the whole practice starting from the Planning, entering the Activity and proceeding with the Evaluation phase. The Activity has been better analyzed and modelled with respect to the other two phases of the practice, which are less structured and comprehensive.

a) Planning

When the therapist plans the therapeutic activity she primarily needs to focus on the therapeutic objectives established together with the other therapists in the integrated rehabilitation plan. She then rehearses the overall picture of the child, focusing on the history of the treatment, the child's previous behavioural answers, their needs and interests. In order to arrange the plan for the session the therapist obviously relies on previously performed successful activities, her personal set of best practices. By identifying play scenarios, along with objects and rules, it becomes possible to create a unique intervention, based on a careful analysis of the child's capabilities and their limits, as well as their preferences and attitudes.

The diverse abilities and personalities of the children have to be specifically taken into account. To avoid imposing their desires on the child, the therapist has to construct the therapeutic situation in a way that it is significant and motivating for that child. Forcing the child into rigid and repetitive activities is the least effective way of engaging the children. On the other hand, neither is it positive to hyper stimulate the child with many different initiatives. They have to be able to understand the requests and to choose the strategies that are most suitable to them in order to carry out the task.

Another element to consider in this phase is the proposed place for all of the activity. The presentation of games and toys and the structure of the activity must also be organized within the swimming pool environment. These choices must match with the purpose of the therapy and the child's individuality. The therapist needs to set up all of the available tools, which must already be configured but should also allow for customization and adaptation. The availability of ready at hand resources contributes to the prevention of distraction and loss of attention and allows the therapist to keep exclusive focus on the patient.

Key issues of the planning are summarized below:

- *Focus on user needs*
- *Focus on acquired objectives*
- *Creative and novel proposals*
- *Re-use of successful solutions*
- *Minding the history of the therapy*
- *Purposeful set-up of the resources*

b) Activity

The play activities must be started, maintained or changed in accordance with the child and their specific way of dealing with the problems and the provocations that the environment and the actors provide her. The play situation is not based on and characterized by the functional limitations and the pathology; rather, it is highly individualized according to these factors, and is based on an extensive knowledge of the child, their preferences, skills and frustrations. In this way the therapist may obtain the connection indispensable to establishing a play relationship with the child. The selected play activity also involves objects that have a role with respect to the exercise and to the relationship. The tools are used to stimulate imagination and curiosity, and involve the child in social and collaborative game dynamics. The activity is highly distinguished by the opportunity to provide the child with free movement in a very pleasing environment. The movement in the water is determined in relation to the treatment, e.g. treating children with attention deficit hyperactivity disorder requires a restrained space of action that does not favor free exploration and initiative taking.

In the following figure a specific model of the activity has been elaborated and evaluated on the basis of the ethnographic analysis and the participation of the therapist in the research process. The model is inspired by and adapted from the Body schemas Intelligent Learning approach (Vayer 1992) that consist of three phases:

- Exploration: Situating the problem and free exploration
- Dissociation: Proposal of different behavioural patterns
- Stabilization: Repetition on interiorization of body schemas

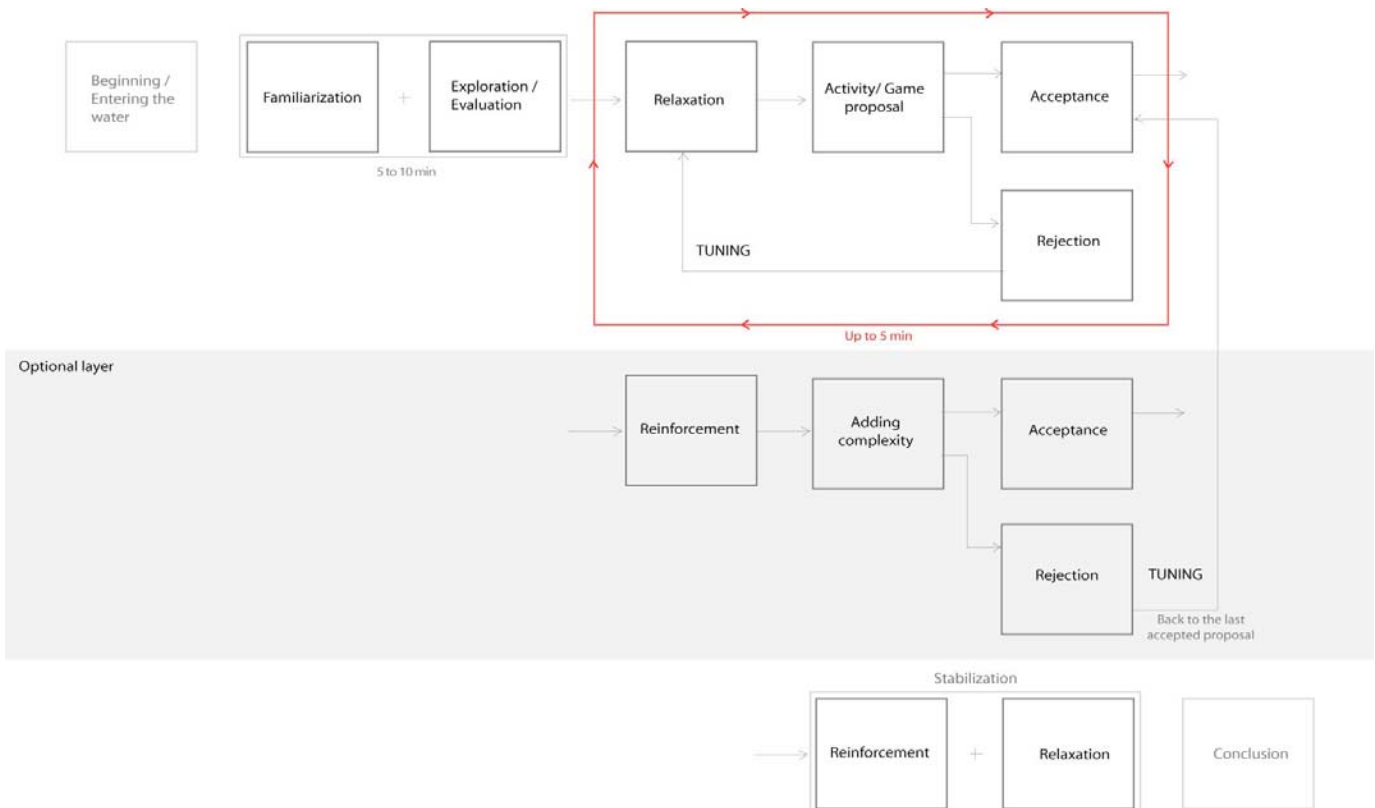


Figure 18. The Activity model

The activity is detailed in the model by a sequence of actions each represented in the squared box. The model shows the relations among the events occurring throughout the therapeutic session. The sequence of actions is described to a large extent as follows.

With the entrance to the pool and the first contact with the water the child begins to familiarize themselves with the novel and uncommon environment of the swimming pool. They must deal with the novelty of meeting the therapist and the extensive experience of entering the water.

This phase is very relevant at the beginning of the therapy. Especially for very young children everything constitutes a novelty, both the unusual bodily experience of the water and the therapeutic play. As they continue with the sessions the children takes less and less time to adapt and to take part in the play activity.

The therapist must then explore the potentials of the intervention, what can be done and what is feasible. They have to evaluate the health conditions of the child, if they are tired or ill. It is also important to contextualize the activity within the events of the week and the daily experiences.

At the beginning of the treatment the exploration takes place almost exclusively on the child, on her abilities and her interests. Then the therapist may also use toys and objects to understand how the child feels, e.g. asking the child to discover objects, to reach and move them.

After these initial phases the therapist tries to relax the child and create the right atmosphere. They can remind the child of the last pleasing experience they have undergone together within the therapy, or they can work on the pleasing effect of the water to start the game activity with a feeling of wellness.

The game proposal is not conceived as a persistent demand that the therapist ought to make. The proposal has to be tailored to user needs and the therapist doesn't have to unreasonably insist in order to avoid the opposite effect, i.e. a rejection. They involve the child in playful situations always *asking* the child for an answer, whether it be the acceptance or the rejection of the game. The child has to adhere to and, in some way, declare their interest in the game. This is necessary for the activity to achieve its goals.

The therapist has to tune the game activity with respect to the child's, often only behavioural, response. They have to reconfigure the tools as well by using them in a newly created way. This is an emergent behaviour throughout the interventions. The therapists need to tailor the stimuli and the feedback with respect to the needs of the children, and also using rigid objects and toys not specifically designed for therapeutic use.

When the game proposal is rejected the therapist has to set all of the tools aside and try to re-establish stability. Time needs to be given (usually 2-3 min) where the therapist steps back and permits the child some breathing space in the session, which is referred to as the "stillness zone" (Brooks, Petersson 2006). This could indicate a preference for the next phase, for example if the child is bored with the choice they rejected it could be a defined pause to reflect upon what had been happening; or it could be an indication of exhaustion and a desire to stop the session.

In these few minutes the therapist may try again following another strategy. After an eventual rejection the therapist ought to go back to a positive situation in order to re-establish a positive attitude. Then the therapist retries the stimulus previously rejected or tries a novel initiative.

Optionally, the therapist may want to add complexity to the activity by adding elements to the game or modifying the activity by using only one object and varying some conditions. It can be proposed within a change in the game or a change of the physical dynamic.

In any case the child doesn't have to lose interest in the object/ game. If this risk is present, it is always better to remove the elements of complexity previously added and go back to the game that was originally established. In this particularly delicate domain the therapist has to be sensitive and able to capture interest and anticipate disinterest.

When the child accepts and participates in the game, the experience of novel capabilities happens with an understanding and awareness of novel spatial, physical and functional relationships. The therapist also aims to reinforce the positive results by repetition and slight variation in the play dynamics.

Towards the end of the session the therapist tries to stabilize the child by providing them with relaxing and pleasing activities. They remove all of the stimuli utilized and set all the objects apart in order to make the environment simpler and plainer. In this phase the objective is mainly to support the proprioceptive experience of the children and to favor the contact with themselves. They exit the pool with a sensation of wellness and a rewarding memory of the experience. It is, in fact, very important to create bridges among the sessions in the pool due to the need for making the unique environment of the water more and more familiar. This phase usually takes from 5 to 10 min.

c) Evaluation

After the activity is completed, the therapist makes a qualitative assessment of the session by considering the particular type of interaction established with the child. The evaluation phase usually takes place in a different environment with respect to the pool. The therapist has gone to the rehabilitation center or home and a certain time has passed.

This phase is not structured according to any procedure. The therapist chooses the suitable ways to organize her notes and comments about the child and the therapy.

The therapists are mainly interested in evaluating the intervention with relation to the therapeutic objectives they have defined in the plan, if and how sub-goals, intermediate and functional goals have been achieved. Another focus of the evaluation is on the child herself and on her attitudes and experience. The play activity is analyzed within the context of the relationship established between the child and the therapist. An exercise may be reproduced during different sessions or with different children, but the relationship within which the exercise and the environments are proposed is unique and built around the therapy. The therapists analyze the roles that have been carried out and how the interaction has evolved.

They also evaluate the methods used during the session and if they were appropriate for attaining the objectives. The therapist indeed chooses tools and games throughout the planning that are then experimented with in the real context of use. She doesn't know a priori if the planned resources would be effective and appropriate. The assessment of each intervention constitutes a piece of knowledge that the therapist has to formalize in order to collect and make these contents shareable. She usually takes a notebook to write down the assessment of each intervention in the water.

Key issues of the evaluation are summarized below:

- *Feasibility of the pursued objectives*
- *Appropriateness of the adopted resources*
- *Adding notes*

2.3.2.1 The pace of the activity and the needs of the actors

Entering into a more detailed description of the *activity* (b), in the following section we discuss the key characteristics of the activity that have triggered the design process: the pace of the activity and the needs of the actors.

Each session in the water lasts 50 minutes with a few minutes provided to enter the water and exit the pool. The general structure of the session is based on the periods of activation/ excitement of the child. In fact, the therapist does recognize the fatigue of the child and the appropriateness of the stimulation, and eventually may choose to relax the child.

We describe a general model for the arousal/ relaxation dynamics that take place into the therapeutic activity by considering the outstanding function of the water and the events hereby described. The therapist allows a recreational pause (10 min ca.) between the two most demanding phases of physical and mental activation/ stimulation (20 min each ca.)

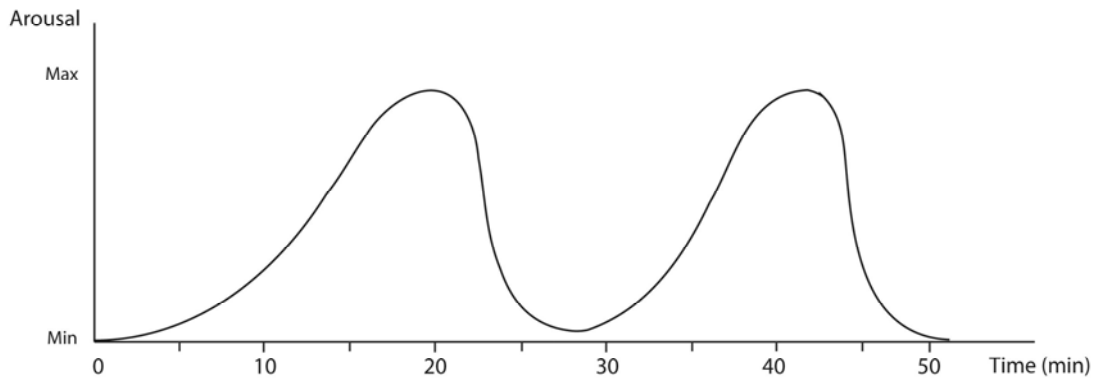


Figure 19. Arousal Model

This graph can serve to model the general behaviour of the therapists and children throughout each session. Of course it may be slightly adapted according to the different patients. For example, in the case of hyperactive children the maximum level of stimulation would be lower than other patients and wouldn't go beyond a certain threshold (Figure 20). In the case of children with severe physical-motor impairments, the activity will probably follow a different path, with a higher periodicity of high arousal and rest (Figure 21).

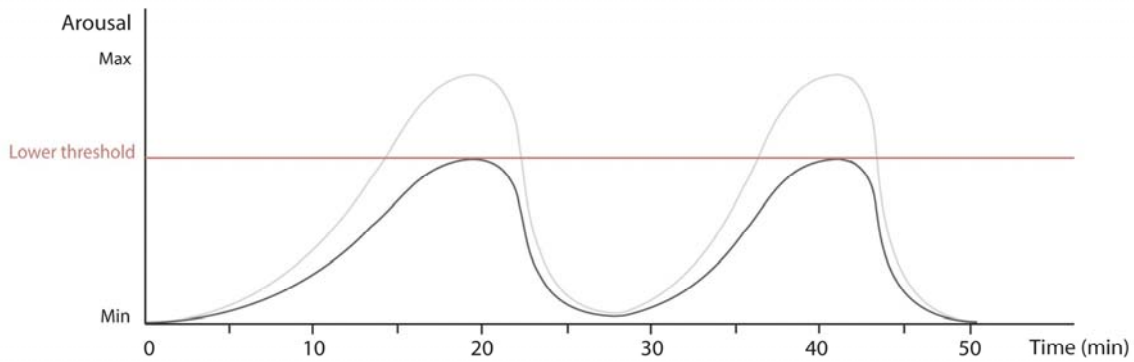


Figure 20. Arousal model representing the case of hyperactive children.

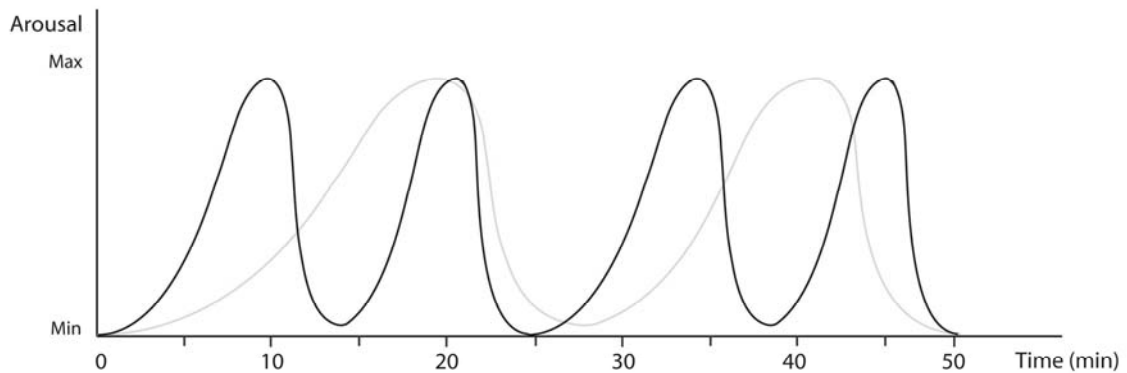


Figure 21. Arousal model representing the case of severe physical-motor impaired children.

A specific pace then characterizes the therapeutic activity in the water. The pace of each session is based mainly on the attention span and on the engagement of the child. The therapy has to remain a pleasing activity that captures and motivate the child to perform demanding tasks while engaging in playful activities.

Generally, children with cognitive deficit and delays are not able to remain focused for more than an average of 5 minutes. Even when involved in purely physical tasks, this special need has to be taken into account. In the figure below we tried to represent the pace of the activity with the same template used in the arousal model. Within the two periods of activation/ stimulation of the child, the activity can't be represented by a linear curve that increases homogenously. The pace of the activity is characterized by 5 minutes intervals during which each proposed game dynamic is maintained and pursued. After no longer than 5 minutes the therapist has to change the stimulus and vary the activity in order to avoid a loss of attention and motivation.

The pace of the activity can be represented with a sinusoid where the games are interrupted by very short intervals during which the child is free to take initiative and thus recover from the efforts made.

During the period of recreation the pace of the activity becomes linear to the lowest level. The overall therapeutic session is thus constituted by up to 10 short game dynamics. It may of course vary depending on the conditions of the children and on how the intervention proceeds.

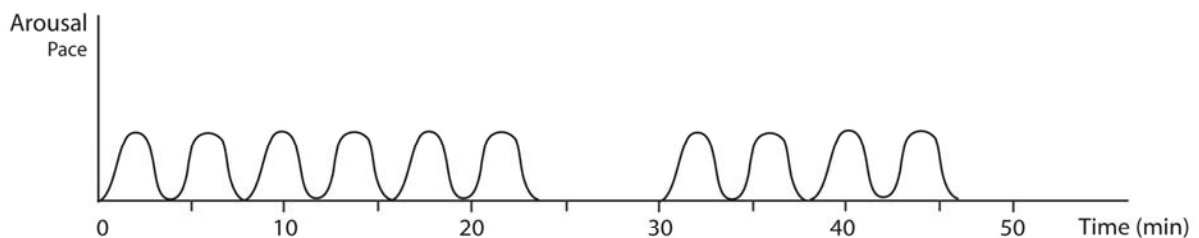


Figure 22. Activity Pace

The models proposed in this paragraph permit a greater comprehension of the current practices and provide a generalization of the knowledge gained from the fieldwork. In this perspective our aim is to pass from the description of single cases to general models through which each intervention can be explained and described.

The game dynamic proposed by the therapists stems from the declared objectives of the therapy (described in 2.1) that can be grouped into areas like Movement, Manipulation, Coordination and Physical Dialogue. Examples of short game dynamics are basic tasks such as following the objects, pushing / pulling, give and take, positioning and orientating objects in the pool, and more complex and structured games where multiple tasks are integrated in meaningful ways. Very often these games are part of playful activities of mutual exchange and reciprocal action.

Key issues of the activity are summarized below:

- *Readily available resources*
- *Exploration*
- *Tuning / adaptation*
- *Pace*
- *Increasing complexity*
- *Emergent use of resources due to changing conditions*

2.4 Activity scenarios

At the beginning of the project, the domain described in this chapter was chosen together with other application sites to explore the PalCom framework, and the choices were based on the consideration and the evaluation of how the scenarios might challenge the conceptual framework.

The domain of the therapy in water was selected because it challenges the PalCom framework in several respects and also presented a strong potential to exemplify a range of practices into which the introduction of palpable technologies may be challenging.

From our field and participatory work we have determined key issues referred the environmental conditions and the users' needs. As described throughout this chapter, the properties of the water enable new opportunities for rehabilitation. The reassuring and calming qualities of the water, as well as the facilitation of body movement that it provides create a pleasant setting in which cognitive tasks can be easily combined with physical ones. The swimming pool represents an environment where different actors own specific spaces and the water play a central role in the activity. The therapists, who are possibly in or near the pool need to re-configure complex assemblies depending on which child they are working with and which support for planned and ad-hoc changes of complex configurations is needed.

The outcomes of the fieldwork and activity analysis that have been described in the previous paragraphs have been summarized and re-elaborated in a number of Activity Scenarios. They have been developed to interpret the user requirements that were determined through the fieldwork activities, and to describe the daily practice of water therapy in a rich, narrative form.

As described in Par. 1.3, the Activity scenarios constitute representations of workflows with specific attention given to the context in which these occur; they describe existing play or therapeutic situations, including the description of goals, artifacts (both material artifacts such as devices, tools, environment and spaces, and conceptual artifacts such as procedures, practices, ethics, etc.) and social relationships aimed at representing how people organize their current activities.

Activity scenarios are the means for representing the current practices and explaining the way in which people handle everyday situations. They stem from the practices outlined above and describe actual examples of usage. By using activity scenarios we aim at describing the experience of the actors involved in the practice and taking in hand their personal and professional experience with target users.

By way of the Activity Scenarios we try to outline key Use Practices. They directly refer to the Architectural Qualities specifically addressed within the Active Surfaces application prototype. A deep understanding of such practices helps designers to explore where and how their work will impact users' activities and informs the software architecture development. The Activity Scenarios have also been used and further elaborated in Prototype Scenarios in the following phase of the process, which will be described in Chapter 4.

The domain of the therapeutic practice in water was selected at the beginning of the project as it seemed to be very promising in challenging the initial conceptual framework and thus guiding the architectural development.

In fact, as described throughout this chapter, the therapeutic practice in water poses challenges due to the changing conditions of the users and especially to the dynamic nature of the context. It has proven to be particularly interesting for the peculiar use and adoption of Ubiquitous Computing technology for therapeutic objectives and towards people with special needs.

Furthermore, from a technological point of view the water challenged the development of embedded, interactive and distributed technologies. Water impacts floating objects by enforcing incessant dynamicity and is particularly challenging to the development of the communication module and the discovery protocol. This has been particularly interesting because of we deal with embedded and distributed Ubiquitous Computing.

The Activity Scenarios described in this paragraph have been elaborated throughout the participatory sessions with the therapists. Specifically, the therapists from the Functional Rehabilitation Unit, Siena Hospital, and from the D. Chiossone Institute, Genova, and the trainers from the Disabled Children Parental Association have been involved in the participatory data sessions. The presence of the stakeholders has been fundamental to begin with a clear understanding of their actual practices.

In these sessions ethnographic data capable of revealing key aspects of user practices have been reviewed and analyzed. In subsequent User Workshops we have modeled the activity and confirmed our understanding of the work, the practices and the tools utilized. User Workshops were essentially discussions, focus groups and brainstorming sessions in which, in turn, the professionals have actively participated as an integrated part of the design team.

The activities addressed by the scenarios strictly refer to:

- the objectives of the water therapy, i.e. movement, manipulation, coordination, and physical dialogue (as they are specified in Par. 2.1.2);
- the relevant issues coming from the therapeutic practice (Par. 2.3);
- the target user groups, with their needs and wishes (Par. 2.1.3).

Objectives, key issues and target users have been articulated in the scenarios to cover all three phases of the activity: the Planning and the Activity in water. The Evaluation phase will be addressed in Chapter 4. The rational behind the scenarios has been outlined in the figure below.

	ACTIVITY SCENARIOS			
	1_ Creatively structuring novel games	2_ Continuous negotiation and adaptation	3_ What's going on	4_ Handling materials
PHASES OF THE ACTIVITY	Planning	Activity	Planning Activity	Activity
ARCHITECTURAL QUALITIES	Assemblability	Adaptability	Resource awareness	Experimentability
Therapist USE PRACTICES	Looking for creative solutions	Dynamic configuration of the tools.	Resource availability and opportunities for action.	Exploration and performance.

Figure 23. The relations the Activity Scenarios have with the Qualities and the key Use Practices.

Activity Scenario 1

Creatively structuring novel games

While planning the therapeutic session for tomorrow, the water therapist, Alessio, sits down in his room and focuses on the therapeutic objectives and the strategies for involving Andrea, an 11 year old boy with severe mental delay and disruption in his orientation functions, i.e. knowing and ascertaining the relationship to objects, self, others, time and space. Andrea has been included in the psychomotor therapy in water for six months.

Andrea really enjoys the bodily experience of being in the water and Alessio usually takes advantage of his positive disposition to the water. Alessio is minding the history of the treatments and the improvements he has observed in the child. Andrea is now familiar with other tools, i.e. the noodles and the tablets, which have been tried out as mediators for social games.

Andrea is now very confident with the therapist and with the water environment, and Alessio is thinking of suggesting novel activities by introducing the sponge noodle, which absorbs and releases the water. The therapist would like to initiate reciprocal exchanges mediated by the use of the sponge, by relying also on the intriguing effect the water has on him. These dynamics may favour the tactile and haptic perception as well as the proprioception. Andrea may be involved with distinguishing differences in material texture, such as dry or wet stimuli, shapes and sizes, and in the awareness of limb positions and body boundaries.

In this way he plans to have the sponge together with the other tools, ready to be introduced in the activity. The sponge could be also divided into two parts if it would help to capture the attention of the child and support imitation games by manipulating pieces of the same material. Imitation may then guide the child into social coordination and may allow the therapist to involve him in physical actions favouring hand and arm use or the use of extremities in order to lift and put the sponge into the water. This can be particularly relevant because Andrea does not always succeed in producing physical behaviour in the proper sequence, such as hand and eye coordination, or gait.

Alessio decides he will start the activity with the tubes by proposing the symbolic game of horse-riding that the child really likes. After having familiarized him with some simple imitation movements, Alessio plans to substitute the tool and, by means of that, he will initiate the novel game activity. In this way Alessio tries to build upon the acquired game schema, e.g. imitation via the symbolic use of the tubes, which has been newly adapted by the replacement of the main object. He will have to continuously negotiate the activity with the child and these ideas will be used as suggestions that would affect the whole session. The ideas here planned will be translated into games through which the therapist will structure the activity, and the effects have to then be entirely explored in practice.

Keywords: Planning, Creative and novel proposal, Purposeful set up of the resources, Newly created formations, Changing an element to provoke a different outcome.

Activity Scenario 2

Continuous negotiation and adaptation

Laura is a swimming trainer specialized for children with special needs. She has been working with Federico, a 7 year old boy with Autistic Spectrum Disorders, for one year. He has severe problems to adapting, acting or reacting to new objects or experiences in an accepting manner or in a resistant manner. His adaptability problems in particular regard the acceptance of activities involving the willingness to act.

Laura meets Federico once a week for 50 minutes sessions. They are now in the water together for a session. They started 10 min ago and Federico is now getting familiar with the context and the situation. From the beginning and throughout the session Laura tries to maintain a dialogue with the child by means of engaging and meaningful games. It is usually very difficult to capture his attention and involve him throughout the activities.

He is now enjoying skimming his hands over the water and playing with the waves. He is very calm and Laura decides to introduce the floating boat toys they have used only once throughout the treatment. Laura puts one boat into the water and pushes it toward the child. After a while, about one minute, Federico starts to push the boat away by producing waves with his hands.

Laura wants to take advantage of his interest in the boat and she begins to move the boat by producing waves as well. They are now moving the boat in a competitive game. As soon as they splash each other, the child starts to cry and shout with discontent: he really dislikes to be splattered with the water.

Now Federico needs time to recover from the stress. Laura is there with him and negotiates between child autonomy and her guidance of the therapy. When the child is calm again he starts to play with his boat. After a while Laura carefully introduces another boat and starts to imitate the movements of the child. Federico starts to enjoy this initiative.

She syncs her behaviour with him and tries to increase the complexity of the activity by starting a more structured coordination game. Each one now has their own boat and manages to exchange it with the other: Federico and Laura push or blow the boats through the waves. This time the child has accepted the proposal of the therapist and she tries to reinforce the stimulus by repetition and imitation. After 3-4 minutes the child slowly loses interest in the game and moves away.

Laura understands that he wants to stay alone and when the child is like that, he usually refuses her invitations to play.

After a minute the trainer tries to approach the child again. She now wants to involve him in buoyancy and proprioception experience. The physical and sensorial experience, Laura suggests, is constantly tailored to the child needs and reactions. She wants to reassure the child and allow him to also gain awareness of his body in the water. Federico hasn't plunged completely into the water yet. He doesn't like to go under the water with his head. While engaged in playing with blow and water bubbles he progressively experiences the water with the head. Suddenly, as soon as he realizes what he has done he immediately stops and moves away. He is now alone again.

Keywords: Tuning; Dealing with uncertainties and changing conditions; Pace; Set-up new games; Diverse needs.

Activity Scenario 3

What's going on

Silvia, the water therapist, is now entering into the water together with Lucia, an 8 year old girl with cerebral palsy. Lucia has been under therapeutic treatment for many years following psychomotor therapy programs. She has currently been undergoing the therapy in water for 2 years. She has problems with her motivation and drive functions, and has weaknesses with curiosity and the desire for mastery challenging and purposeful tasks.

While planning the session for today, Silvia wanted to assess the Hydro Belt bought some days ago and determine if it would fit with the particular demands of her patient. Lucia needs to be sustained but suffers from rigid aiding tools.



Figure 24. The OKEO© Flexible Hydro Belt taken as inspiration.

Silvia handled the Hydro Belt and tried to figure out how to utilize it. She wanted to test the opportunities it offers and understand how the child would behave when sustained by the belt. Before introducing the new tool into the activity, Silvia needs to feel in control of the overall setting, i.e. the environment and the tools. The therapist has it in mind to take advantage of the novel posture that the belt allows in order to involve the child in both cognitive and physical activities regarding the use of the hands.

When the session begins they are both in the pool and Silvia starts with a few exercises to relax the muscular tone. The child is taut and tense and really needs to calm down and relax. Silvia holds the child by the ankles and the back and moves her in the water to stretch and release the muscles in a gentle way.

After a few minutes Lucia becomes very calm, she is laughing with the therapist and seems ready now to attempt some more structured and demanding activities.

Silvia decides to use the Hydro Belt. While holding the child with one arm, she takes the belt that she has positioned on the edge of the pool. Now just the belt sustains the body of the child because the other aids they are currently using have been removed. The child is permitted to stay in a vertical position as if she was sitting in a chair. In this way Lucia can use both of the child's arms and she gains many more opportunities for action.

Silvia starts to use a ball to mediate the interaction with the child. They pass the ball back and forth between each other and Lucia is really enjoying it. While moving, the child progressively loses her posture because of a loosening in the belt. Lucia is now getting very nervous because she feels like she's losing her balance.

Silvia understands what is happening and immediately tries to recover by holding the child and adjusting and fastening the belt in the correct way. Even though it was closed, the belt wasn't previously fixed in the correct position on the discs. This was the first time the Hydro Belt has been used and Silvia has to understand how it might be used in her future practices.

Keywords: Available resources; Available opportunities; Control over the response of the child and the overall situation (environment, other actors, tools); Acceptance and rejection and evaluation of the available resources; Ready at hand resources; Exploration.

Activity Scenario 4

Handling Materials

Elena is a novice psychomotor therapist who has been approaching the water for few months. She likes to adopt different materials and tools throughout the activity; she believes that they can be really helpful to sustain her practice. Elena is now using the set of noodles and connectors together with the snake lane for the intervention with Enrico, a 10 year old child with a mild mental delay. The child really likes playing social games and being involved in symbolic play and storytelling. Enrico has a disruption of control producing psychomotor retardation (moving and speaking slowly; decrease in gesturing and spontaneity) and his motor impairment prevents him from having strong physical activity.

Elena works in both the directions of social coordination and physical activity. Throughout the session she explores different formations of the noodles, utilizing them as if they were bridges, hooks and ropes or, connected in several ways, as a cage or a car. The symbolic use of these components is based on the possible shapes and combinations they allow, e.g. by the use of four and six holes connectors. While having this set of heterogeneous components, Elena also needs to understand if her inventions are robust enough to be used with the child. Before the session Elena has only planned the kind of games they will play, and then during the session she has to physically explore the set of tools in practice.

They are now in the water; she holds the eight-hole connector and, together with the help of the child, she is constructing a spider. Enrico holds the six flexible legs and puts them into the connector, i.e. the body of the spider. Two out of the six noodles can't be attached in such a way that they respect the shape and the behaviour of the spider. In fact, in the set of tools, only four short noodles are universal, the rest can be used only in specific ways.

Elena wasn't aware of this particular restriction of the tools until she tried them out. Until she is able to handle the materials in the actual context of use, she has little idea of the strengths and the weaknesses they might have. Therefore, she is now experimenting with the noodles again to quickly determine a new game that would attract the child in the same way.

Keywords: Handling resources; Performance; Error tolerance; Try and error; Exploration and evaluation; Emergent uses.

These Activity Scenarios specify the therapists' Use Practices that have been explored throughout the research in a concrete context of use. The following key practices,

- Looking for creative solutions,
- Dynamic configuration of the tools,
- Resource availability and opportunities for action,
- Exploration and performance,

were among the outcomes of the initial exploration of the application sites (as reported also in PalCom Deliverable 12) and have continuously informed the development of the software architecture. In fact, the investigation of such practices made the enabling architecture necessary to realize the vision of *palpable* Ubiquitous Computing.

In fact, the exploration of actual activities and Use Practices has direct correlations with the Architectural Qualities. This issue can be described as a continuous dialogue between the software architecture and the use. The exchanges between these two domains have been made possible by way of the Activity, Envisioning, Prototype and Qualities Scenarios that bridge the therapist use practices to the implementation

of the architectural qualities, and vice versa.

By means of the Activity Scenarios we tried to understand, as thoroughly as possible, what is relevant and appropriate in the specific domains of use, which in this case study was the therapeutic practice in the water. We described how users engage with the resources available within the environment and we use this information to inform the whole design process.

The key therapeutic Practices were also described in par. 1.2 in a more generic way. In the Activity Scenarios they are explored, specified and explained in great detail. The therapeutic Practices have been re-formulated with the active participation of the users. They can be summarized as follows:

Looking for creative solutions: The therapists usually deal with dynamic settings and changing conditions. This implies the ability to manage and rearrange the available resources in purposeful and creative ways. This practice is described in Scenario 1: “Creatively structuring novel games”.

Dynamic configuration of the tools: In dealing with continuously changing conditions and rehabilitation demands, the therapists should always find new solutions for adapting their tools and the environment to the patients and for maintaining their attention throughout the session. Consequently a core characteristic is that the tools have to be easily re-configurable and adaptable to this evolving situation. This Use practice is described in Scenario 2: “Continuous negotiation and adaptation”.

Resource availability and opportunities for action: The therapist needs to feel in control of the available resources and how they might be adopted, changed and exploited. As in many workplaces, since their attention is exclusively directed to the patients, the resources the therapists use have to be ready at hand and immediately understandable. This practice is described in Scenario 3: “What’s going on”.

Exploration and performance: This practice facilitates and encourages exploratory experimentation by users. Tools have to be used, customized and altered according to established degrees of freedom and constraints. This practice is described in Scenario 4: “Handling materials”.

Chapter 3

Active Surfaces: Design and Development

The iterative process, described in Par. 1.3, sets the concept design and development as one of the activities that have been carried out simultaneously with fieldwork and activity analysis.

In this chapter the generation and development of the Active Surfaces concept will be described. The approach followed in this research, the Therapeutic Play (Par. 3.1), the concept itself (Par. 3.2), the concept evaluation and refinement (Par. 3.3) and the development of the concept (Par. 3.4) will be described. The outcomes of the concept development will be summarized and further elaborated in the Envisioning Scenarios (Par. 3.5).

3.1 Therapeutic Play

Throughout the fieldwork and activity analysis we have gained inspiration for the kinds of activity to be supported. Two complementary investigations on the approaches for water therapy (Par. 2.1.4) and on the current practices of the therapists (partly described in Par. 2.3), gave us the ideas to explore the concept.

The main approaches that address the therapeutic practices of disabled children are the Psychomotor Therapy in water, a relational approach characterized by exploration, symbolic play and play activities, and the Aquatic Therapy, which is mainly characterized by structured programs of exercise and procedures.

In the design process we tried to develop an intermediate and integrated approach out of these well consolidated two. With this approach, the therapeutic play (Piaget, 1962; Vygotsky, 1966, 1997), we aim at combining cognitive and motor-physical rehabilitation.

The lack of integration between cognitive and physical rehabilitation represents a relevant aspect in the current rehabilitation practice. The cognitive tasks are usually quite boring for the children and they quickly get tired and lose attention. On the other hand, motor rehabilitation is very demanding at a physical level and is based on repetitive sequences of actions often perceived as very tiring and not engaging at all. By merging the strengths of the two approaches it may provide the patients with the motivation for demanding activities and the therapists with flexible and adaptable tools for tuning and controlling the activities.

The Therapeutic Play as explored in this research can be considered a part of the psychomotor therapy approach, particularly close to the Body Schemas Intelligent learning (Vayer 1971) and the Psychomotor education (Le Boulch 1975) (Par. 2.1.4.1). The hypothesis behind this proposal is that in the context of the therapy in water, activities in which rehabilitative tasks are proposed to the child in a play-oriented and a possibly fun environment are also possible.

The proposed approach has a precise therapeutic purpose and has a play-like nature; both of the two aspects are balanced and emerge in equal parts. If the exercise aspect dominates the child may not be totally involved and motivated in the activity. On the other hand, if the play aspect is too overwhelming, the activity may lose its efficacy towards therapeutic objectives. In fact, the therapy is strengthened by defining play-based functional movements associated with cognitive tasks, in order to improve motion as well as the cognitive skills associated with learning and play (Bruner 1992). The therapist is the play partner who can offer those enrichments and scaffold the development of the child. In this way the therapist can grab the attention and the interest of the child and maintain orientation toward objectives.

The subjects can experience with their potentials and limits through the meaningful interactions they have within the physical environment with objects and with other subjects. In this account the tools are viewed as integrated into our body because of the potential for action they provide (cfr. the *bodily space* in Merleau-Ponty 1962). The tools used in the therapy may also enhance the physical agency and the opportunities children have to act and react within the environment, and this especially happen in the water.

Together with the opportunities it provides, the adoption of tools is also linked to how much the activity they support stimulates motivation and participation and is appropriate for the subject's physical and

cognitive level (Besio 2003). The motivational aspect plays a fundamental role if we want to maintain involvement in the activity and achieve the established objectives. The ways in which an adequate level of motivation is obtained vary significantly depending on the individual, and a play-like activity helps to create those motivational aspects (see Par. 2.1.1).

3.2 The concept

The Active Surfaces concept was generated in a preliminary formulation within a master students design contest, the Siena Design Project (SDP) 2004. The early concept was then refined and developed throughout the design process in the PalCom Project. It has been elaborated by means of different methods, like the attribute listing or the participatory evaluations with our stakeholders, in order to allow the early conceptual intuition to evolve into a mature concept.

The main idea of Active Surfaces is to rethink the environment of the pool, making it a place for rehabilitation and play activities. Indeed, the swimming pool is conceived as place to swim, and other kinds of activities that may take place in the pool, like the rehabilitation activities, are not currently purposely taken in hand. Active Surfaces re-considers and re-interprets the environment of the swimming pool by focusing on those surfaces that constitute the physical environment: the bottom, the water surface and the vertical walls, in order to support Therapeutic Play activities.

Active Surfaces is a modular system constituted by physical and interactive units, the tiles. They are interactive modules that activate the surfaces of the swimming pool by making the environment featured with a network of distributed interactive components (Gronvall et al. 2006a). In particular, the concept as developed in this research affords the horizontal configuration of the tiles on the water surface.

Active Surfaces accounts for the specific needs of the therapeutic practice in the water by embodying the following features:

- * configurability,
- * constructability,
- * modularity,
- * physicality,
- * support for creativity.

These attributes are described in the following paragraphs as being embedded in design and architectural solutions adopted for the development of Active Surfaces.

The Active Surfaces system is designed to be intuitive, engaging and accessible to the users, including both the therapists and the children with special needs. It should provide a rich and flexible terrain for the users to explore the possibilities offered by the architecture.

The tiles are tangible objects that can be controlled by physical body movement, by appropriate turning, moving, and manipulating. Active Surfaces is an example of computational technology that is physically embedded in a congenial way for users with diverse competences and abilities. The Active Surfaces concept represents the idea of making those physical resources digital and interactive, with the aim of allowing the configuration, adaptation and goal directedness of the tiles' physical modular system.

The tiles constitute a network of physical (and software) objects that communicate and exchange data. In addition there is a privileged tile: the Assembler Tile, which is used by the therapists to program the other tiles. A number of tile components can be logically assembled in ad hoc formations using the Assembler

Tile and can constitute a network of physical (and software) objects that communicate and exchange data. Each Active Surfaces tile is thought of as a modular unit that can communicate with the others through its six sides. The tiles are, in fact, able to recognize their relative position as being essentially positioned and orientated in a sequence of tiles.

Therapists can define rehabilitation activities by configuring tiles and the patients can place the tiles in meaningful combinations. The tiles act as building blocks that can be combined with different content, such as images, sounds and pictures.

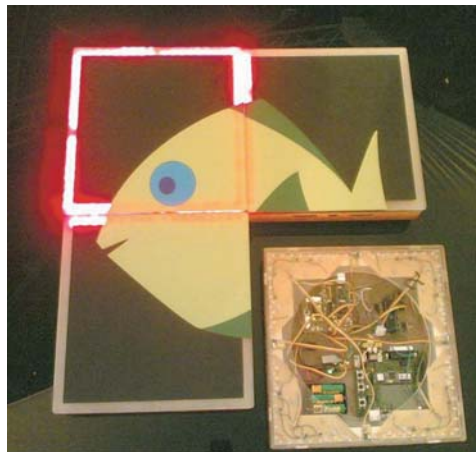


Figure 25. The current Active Surfaces prototype

Physical positioning is among the main features of the tile components. The Tiles recognize their location and functionality can also be triggered by the construction of spatial and logical relationships. The distribution of the modules throughout the space may support the exploration of the environment and the acquisition of spatial- temporal coordination.

Active Surfaces constitute a physically and logically configurable assembly in which it is essential that each tile preserves its ID in every configuration (Pollini, Grönvall 2006). Maintaining the identity is a condition that allows the persistence of the system and creative change in the tile configuration: while the tools will allow great flexibility, they should guarantee the stability and continuity of an interaction. Furthermore, the tiles may have reactive and proactive behaviours in relation to environmental changes or different input actions. Each tile may provide visual, acoustic or tactile feedback to support the accomplishment of the tasks. The Active Surfaces have a high combinatory power and may suggest new emergent activities to the therapists. It is a flexible tool that also allows the patients to behave creatively, e.g. associating functions and services by combining the tiles-components. At the same time, the tiles need to communicate what they can do very clearly to the users, especially their connectivity. Because they are conceived as tools for the water environment, the tiles need to guarantee continuous connectivity within the assembled configuration.

The Active Surfaces is highly scalable in respect to computational power and number of components. In fact it can scale up or down (vertically) by adding or removing resources to a single node in a system, typically involving the addition or removal of CPUs or memory to a single tile. Active Surfaces can also scale out (horizontally) by the addition of more nodes to a system, such as adding new tiles to the distributed system.

The strategies of interaction with Active Surfaces were explored through empirical investigation, mainly by studying the physical arrangements of the tiles, how users behaved in physical and spatial arrangements and how the different contexts provide us with opportunities for action.

3.3 Concept evaluation and refinement

The Active Surfaces concept has been evaluated and refined through iterative participatory sessions. Throughout the different exploratory activities we evaluated the concept with the stakeholders in order to better understand their work practices, their goals and, moreover, the possible uses of the final system.

Several activities have been carried out respectively through the early proof-of-concept and the late concept development. The evaluation of the concept resulted from the number of different activities described below, part of which is also described in PalCom Deliverable 33.

- *Explorations with the Java PoolSim application.* Throughout the proof-of-concept we started to develop the PoolSim Java application to give shape to the concept and have something concrete in order to convey the main idea to therapists. The PoolSim materializes basic features and functionalities of the envisioned system. The tiles have been represented as movable squares of different colours that recognize their relative position and, while in the vicinity, react by changing colour and blinking. The PoolSim has been used as a tool to allow the stakeholders to try the system as it has initially been conceived and to support the emergence of new topics.

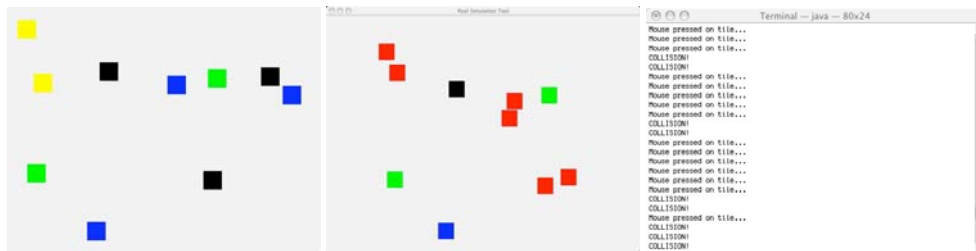


Figure 26. Screenshots from the PoolSim Java application

- *Concept evaluation workshop.* In different meetings at the hospital we began to envision features of the application such as input and output systems, communication modalities and configurability. We addressed these issues by attempting to figure out the needs of the stakeholders and asking for comments and suggestions. In particular, the system's configurability was one of the most valuable properties of the Active Surfaces concept. The therapists need to adapt the system in order to design as many activities as the individual needs of the children require. Active Surfaces needs to be a flexible and configurable technology to be used for different therapeutic objectives. Therapists suggested many meaningful uses by envisioning both physical features (e.g. how to snap the tiles) and the functionalities of the system.
- *Wizard of Oz in the swimming pool.* The Active Surfaces concept was also put through a number of Wizard of Oz sessions and the outcomes from these activities have informed the construction of the interactive mock-up and prototypes. We have adopted two Wizard of Oz sessions in both of the settings, the gym at the hospital and the swimming pool. We aimed at testing features of the tiles (like their physical properties) with users, following the scenarios we developed together with our stakeholders. In the Wizard of Oz method, the experimenter manipulates the interfaces of the system and reproduces what the system functionalities would be. He makes commands happen and behaves as the actual working component should. Relevant issues in designing Wizard of Oz

sessions are the nature of the used technology (functional, low-tech or non-tech prototypes), the visibility and the knowledge of the wizard (he may be seen or unseen, he may see everything or partially), and user knowledge and understanding (in relation to the wizard's visibility) (PalCom Dev. 33). Different environments were addressed both throughout the SDP04 student contest and in further investigations within the project. In the former case the primary environment was the swimming pool, while in the latter it was the gym at the hospital.

In the swimming pool the children were engaged in a *follow-the-path* task in both horizontal and vertical configurations. The follow-the-path task is based on guidance from the therapists and feedback from the system (played by the wizard). During the Wizard of Oz sessions in the pool several wizards played auditory feedbacks from outside the water while other researchers, acting as therapists, assisted patients in accomplishing the tasks. For example, in the horizontal configurations the tiles were placed on a transparent surface on which they were fixed in the same positions. We used low fidelity mock ups that embodied some physical and material features of the tool. In this way we assessed properties like weight, size, materials and possible uses. Patients had to follow a path made up of similarly colored tiles or follow a previously shown model. The outcomes of these trials can be summarized as follows:

- Horizontal configuration stimulated movement in motor impaired patients,
- The tiles enhanced socialization and attention,
- Pre-defined paths didn't fit the needs of Down Syndrome patients,
- Vertical configurations represented a more difficult task (learning to dive),
- Problems related to acoustic feedback: the noisy environment and hearing sounds under the water



Figure 27. The Wizard of Oz sessions at the swimming pool

The Wizard of Oz session performed in the gym at the Functional Rehabilitation Unit was more structured than the previous one described above. It was based on cognitive tasks involving both the two configurations. We set up the equipment in a room behind a one-way mirror where wizards simulated the behaviour of two tiles through the use of laptops and flat screens. On one side of the mirror we attached coloured foam frames around the displayed picture to evoke the image of tiles familiar to the children. Furthermore, the frames were taken from the same tiles used for the horizontal configuration. There were 8 tiles on the ground (4 orange and 4 blu) which provided users with auditory outputs thanks to the wizard hidden in the gym.

We tested cognitive tasks with autistic children and twins affected by Angelman syndrome. Angelman syndrome is a very rare and severe cognitive and physical disease, which affects motor coordination, language development and cognitive skills. The *image-matching* and the *following the path* tasks were proposed to both of the testers. To complete the image-matching task, patients

had to select pairs of matching images by touching the ‘tiles’ on the mirror. One image was still and served as model for the selection among four different images displayed on the other ‘tile’.

The *following the path* task is based on the use of tiles of the same colour. We also tried group activities, asking testers for following different colours and guiding them with different auditory feedbacks. The Wizard of Oz tests spotlighted important suggestions related to individual and collaborative task design and also more specific features such as providing acoustic or visual feedback on users’ actions. The combination of acoustic, tactile and visual stimuli seems to be a key element in engaging the users in complex activities.

The different configurations widened the design concept, thus allowing us to evaluate the discussed games and activities and compare them with specific cognitive tasks that are meaningful for therapeutic objectives.

The outcomes of these trials can be summarized as follows:

- Material properties of the tiles: the soft frame of the tiles was more attractive than the picture as such.
- Meaningfulness of acoustic feedback: spatial source of the sounds, volume, coherence and positive/ negative differences.
- Semantic matching task: depending on the cognitive abilities of patients.
- Iconic Vs figurative representations.



Figure 28. Wizard of Oz sessions. On the left the trial with an autistic child, on the right the trials with twins affect by Angelman Syndrome.

The outcomes of these activities are hereby described in this section. In the meetings at the hospital we began to envision features of the application such as input and output systems, communication modalities and opportunities for configurability. Then we addressed these issues and attempted to determine the needs of the stakeholders by means of the descriptions provided by the scenarios of use. On the basis of these scenarios we once again received comments and suggestions for the future iteration of the design attributes of the Active Surfaces and also on how they can be achieved with the software architecture.

The main qualities of the concept can be summarized in the manipulation and tactile experience, the possibility of allowing horizontal and vertical activities, the display of pictures and letters, the different configurations of assembly (both lateral and tower case assemblies), flexibility and understandability and the communication and positioning capabilities.

Active Surfaces are essentially mobile, hand-scaled devices, which is why we also consider issues related to the direct manipulation of computational resources and the way in which this affects both the tiles affordances and the manipulation abilities of the users. In fact, the movement and the high versatility of our hands are the underlying skills for the activity with Active surfaces, since it is a distributed, embedded modular system. Both gross and fine motor manipulation are thus critical to understanding the needs and

constraints involved in the design of handheld compositional devices, especially when designing for people with special needs, who do not have fully developed manipulation abilities. Moreover, in our scenario potential instabilities and perturbations caused by the water may occur throughout the activity. The dynamic movement of the objects in the water implies also the dynamic grasp and a very focused attention on such moving objects.

Even if the main inspiration for this concept comes from the rehabilitation activities performed in the swimming pool, wherein water represents a stimulating and safe setting, the system would also be used for rehabilitation activities in the gym. In fact, Active Surfaces can be dynamically configured and assembled by the therapists according to the needs of different patients, the usage, the physical space and the required complexity of the situation.

As outcomes of the concept evaluation we have focused on specific therapeutic and play objectives. Active Surfaces has been used essentially for assemblage problems. The modules are used to build assemblages or sequences and the game implies the re-construction of the physical and logical arrangements of the pieces mainly through two different steps: the *selection problem* - which piece is to be attached to the next - and the *placement problem* - where and in what way should it be attached. The system is designed to support the construction of the sequence and to provide feedback throughout the activity. In particular, due to their physical appearance and interactive behaviour, the tiles can be used to explore therapeutic objectives like spatial orientation and spatial reasoning, sequence learning (and procedural memory), gross-motor movement, socialization for group activity.

These objectives served as inspiration for the further development of the application and the design of the play activities. The overall goal of the activities with Active Surfaces is to provide the children with a playful experience that has therapeutic purpose, that support sense making and that is appropriate for the special needs and abilities of disabled children.

The knowledge produced in this phase has informed the following development of the concept throughout the prototyping phase.

3.4 Concept Development

The outcomes of these initial concept evaluation activities have informed the iterative process throughout the concept development. In fact, after the concept was presented and discussed with the stakeholders, we received meaningful feedback that helped us to refine some aspects of the concept and to evolve it into a mature working system.

Throughout the concept development iteration we carried out the activities described below, the outcomes of which have informed the further prototyping of Active Surfaces.

- *Design workshops.* Different design workshops have been carried out in order to develop specific parts of the concept and to prepare for the subsequent development of Active Surfaces. The first workshop concerned the early implementation of the Tiles and of the functionalities they must have. Architecture developers from the Computer Science Dpt., University of Aarhus, designers and therapists worked together and played with the toy tiles that were purposely built to stimulate the design process. We reflected upon theoretical and technological challenges. We identified specific challenges related to this application prototype. They constituted major opportunities for designing support for therapeutic activities in the swimming pool. The challenges can be grouped into issues such as interaction modalities and physical properties of the system. Despite this categorization the issues are strictly connected to each other. In the interaction with the tiles both

therapists and patients performed different behaviours, on one side through the programming and shaping of the connections, on the other by manipulating, moving and combining them. The use of the tiles involves relevant features such as recognizing whether they are connected or not, the directions of connection, IDs, and inclinations. We discussed what level of interactivity they would allow and how many possible configurations they are designed for. The therapist in our scenario can configure the tiles for the activity by interacting with a GUI connected to the assembler tile. She can select the behaviour (service) to be downloaded to the assembler and through it the service is transferred to each tile in the proper pattern, giving the correct ID and behaviour to each. The configuration can occur also in the opposite direction and the therapist can record the physical configuration she wants to work with on the assembler. The therapist can now program services on the saved pattern. We identified the requirements related to the tile assembly as the stable and flexible connections, the discovery of devices with meaningful identity and the saving and loading of different games.

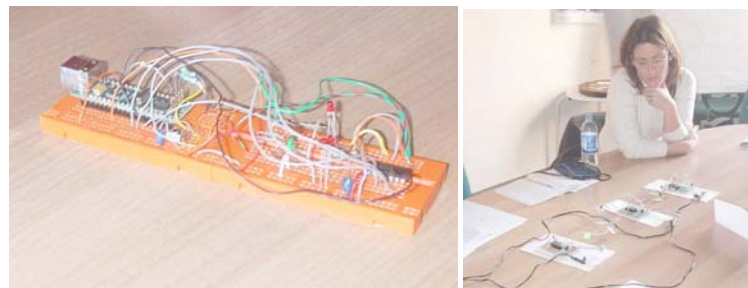


Figure 29. On the left, the early implementation of the Tile concept. On the right, the therapist playing with the three toy tiles along the Design Workshop on tiles and functionalities.

The second Design Workshop was particularly concerned with the initial design sketches of the concepts of the Active Surfaces and how they could be implemented in the physical design. Therapists from the Disabled Children Parental Association and partner industrial designers from the Aarhus School of Architecture worked together with the design team to define the ideal types of games and feedbacks; they maintained strong focus on user needs, eg. what physical shape they could take and what games must be configured. The design sketches and mock-up supported our discussion on fundamental design issues. Some of them are herein outlined as examples of the kind of activity carried out in the workshop. We evaluated issues from the different curves of the bottom of the tiles and the related safety issues, to how to make the tile a floating waterproof object. It was also important to place the communication port, the magnets for the attachment and evaluate the possibility of having LEDs.



Figure 30. The tiles take form in the initial design sketches and mock-up produced by the Aarhus School of Architecture.

The shape of the tiles depended on their intended function and their ultimate environment. If they are going to float around in the water, the shape of the tiles could easily be inspired by a ship, or if children will walk on them in the gym they should probably resemble something you could step on, e.g. the tiles in the pavement.

The size of the tiles then depends on the angle from which the user will see the tiles. If they use the tiles from a standing position, the tiles can be much larger than if they see the tiles at almost eye level. In empirical tests we found out that each tile could reasonably measure 30*30 cm. This standard tile could also be easily handled with two hands. This issue is also related to the visual feedback, the visibility of the lights from the child's angle and how can we design different feedback possibilities and test them in water.

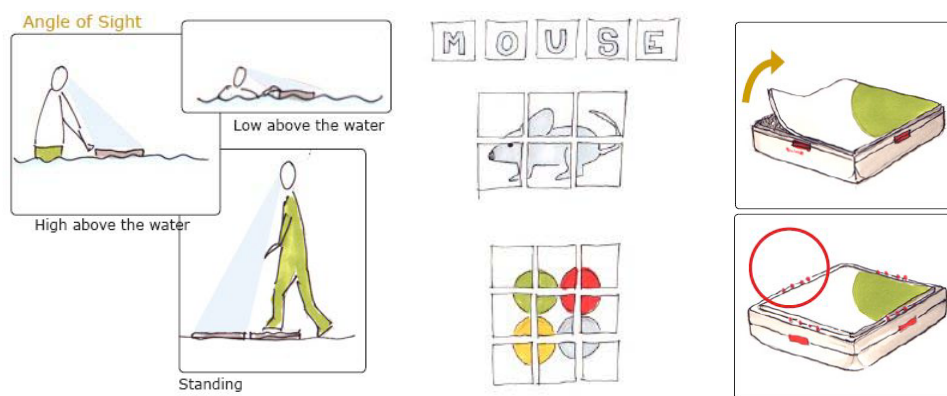


Figure 31. On the left, the angle of sight, on the middle, the types of game, on the right, the solutions for applying the surfaces and positioning the LEDs.

In this workshop we brainstormed and evaluated the different design solutions and discussed new potentials and design opportunities, referring to issues such as Feedback, Lights, Form, Construction, Games, Attachments and Tactility.

The criteria that inspired the process where:

- Flexibility: being able to configure many kinds of games with the possibility of work both in the gym, and under water.
- Simplicity: both of the game and the tile surface. It should be very easy for the children to understand the game and to change the interface of the tile.
- User-friendly: easy to understand the meaning of the tile for both the therapist and the children.
- Tactile surfaces: something that the child can experience and sense.

Throughout the workshop we also performed trials in the pool in order to review the context together and demonstrate the initial lightweight prototype we developed on site. The initial lightweight prototype is described in the following paragraphs.

- *Travelling Architects Workshop.* Throughout the Travelling Architects (TA) workshop a group of architects from the Computer Science Dpt. University of Aarhus take turns in travelling to the different partners' site to learn from their architectural ideas and inform the PalCom architecture (PalCom Deliverable 33). Additionally they bring with them the ideas already defined to help the different partners design their software with the common architecture in mind. Travelling architects participate in some fieldwork, participatory design and evaluation activities and play a

key role in connecting use, application prototypes and architecture.

The TA is primarily conceived as a technical meeting, but it has also been the main way to keep the architecture aligned with the requirements of the prototypes since the decisions about the architecture at the same time affect the prototype's features and functionality and receive contribution from an effective understanding of the scenarios / use domains.

Throughout the TA workshop the Active Surfaces vision, ideas and cases have been presented and the progress of the current implementation has been discussed and reviewed.

Then designers and travelling architects (i.e. software architecture developers) went on documenting the architecture of the current implementation in order to detail what has been done and what was still missing to have a fully palpable tiles system. Then we also prioritized the Active Surfaces architectural Qualities by going through a set of scenarios to understand which Qualities are relevant and what are the implications for design. The goal of this activity was to start thinking about which qualities are important and what are the necessary steps to take them in the development of the architecture.

We then began discussing the construction games envisioned with the therapists in the previous design activities and focused particularly on Position games and Scrabble-like games. We identified issues concerning rotation and interchangeable positions among tiles.

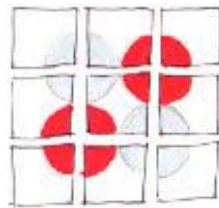


Figure 32. Pattern that present interchangeable positions among the tiles.

The first one is related to the patterns that present interchangeable tiles, which have exactly the same surface or can be placed in the same position if appropriately mirrored. In this case by either permitting or forbidding the rotation or mirroring of the pre-programmed “correct positions” presented a problem. There is in fact a need to enable surfaces to be interchanged based upon the pattern or figure.

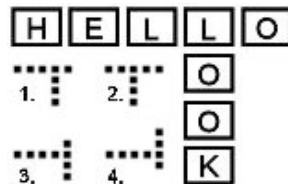


Figure 33. Possible emerging uses.

A related issue arises from emergent uses and unpredicted and un-designed solutions. In fact, as represented in the figure above, we also have to consider what happens if the child composes HOOK instead of LOOK by placing “O”, “O”, “K” in a row with “H”. In fact, Hook is another sensible solution for a child. In the workshop we discussed how to manage these kinds of emerging challenges. Other surfaces, like “O” in our example, can be rotated throughout the activity while preserving the same meaning for the user. If we choose to use these kinds of surfaces, the rotation of the internal pre-programmed “correct position” needs to be allowed.

Based on such kind of discussion, the Travelling Architect meeting resulted in a specification on

the palpable aspects of the Active Surfaces application. We tried to make the application prototype design fit into the current implementation of PalCom Open Architecture, basically through the PalCom ontology, and to give indications as to where the architecture need to focus, by providing feedback about either acceptance or rejections on changes to the architecture.

- *Game Logics workshop*. Developers from the Computer Science Dpt., University of Aarhus and from the Computer Science Dpt., Lund University met the designers to work on the tiles game logics. Throughout this workshop we formulated the concept of Distributed Happiness for defining the game logics in Active Surfaces and the implications it might have for software architecture. The games we have been designing with Active Surfaces follow a logic based on condition satisfaction rules. Tiles' states are described through the use of a "happiness" state. These terms are used with specific meanings in the scenario and in the code development (Gronvall et al. 2006b). We consider different states of happiness (conditions' satisfaction) for the position and orientation of the tiles in the assembly.
 - *SideHappiness* means that a tile realizes that it is correctly connected on a particular side. On the side(s) that is Happy the tile provides the users with HappySide feedback. If, rather, all its sides are correctly aligned, *LocalHappiness* is achieved.
 - *LocalHappiness* means that the tile is properly connected to the others and it has the tiles that it was looking for on each side. It is in the right position and it is correctly orientated in the assembly.
 - *UnHappy* . A tile is unhappy when it lacks *LocalHappiness* (i.e. while no happiness state or only *SideHappy* is reached, a tile is still *UnHappy*). While a tile is *UnHappy* it broadcasts its *UnHappy* state to the other tiles.
 - *AllHappiness* means that all the tiles satisfy the *LocalHappiness* state. A tile that is not *UnHappy*, does not communicate anything to the other tiles. The lack of *UnHappy* messages (given a certain timeout) within the system together with *LocalHappiness* realizes the *AllHappiness* state. The game is thus solved.

We discussed the broadcasting of the GameRuleSet and in particular what happens when a new game is instantiated in a pattern and Tile 1 has to broadcast the GameRuleSet to all the others. It became critical to understand how long the microprocessors would take and how the PalCom architecture could support such a configuration of the game. It was envisioned that the tiles ought to have different functioning modes as it is in the following example:

- the programming mode
- the playing mode
- the inspection mode

The Envisioning scenario described below (Par. 3.1.5) represents and summarizes the discussion and the outcomes of the Game Logics workshop because it describes the future use of Active Surfaces from a logical perspective (Gronvall et al. 2006b).

The activities that have been carried out throughout the concept development also resulted in the definition of different games to be supported with Active Surfaces. The games are mainly based on construction of assemblies of tiles and distant or contingent communication among them.

In all the designed games we deal with assemblage problems that have the selection part, which piece is the one to be attach next, and the placement part, where and in what orientation should be attached. In fact each game is characterized by the possibilities each tile can have in an arrangement; in some games there is unique position and orientation in other two or many possible placements.

In complex task with a set of many tiles, the orderings can be more elaborated. An example of the most common strategies to solve assemblage problems is creating sub-assemblies of tiles and then assembling those sub-assemblies (Kirsh 1995a). The structural properties of the concept permit to create games and tasks with different complexity levels, both by the addition of tiles components and the arrangement of the system in the space.

The kinds of game activities designed with the therapists fall into four groups:

- Jigsaw Puzzle (e.g. the picture of a fish)
- Scrabble (e.g. with the letters “G”, “O”, “A”, “L”)
- Domino
- Ordering task (e.g. ordering circles from the smallest to the biggest)

As examples the Jigsaw puzzle game and the Scrabble game have been hereby described as follows.

Jigsaw puzzle game

In solving jigsaw puzzle the recurrent problem is to identify, from a large set of pieces, the single one that correctly fits the target space. One of the properties of the game is that both fine and coarse discriminations are necessary, especially in puzzles with many pieces. Coarse discriminations help with planning and fine discriminations are necessary to determine exactly which individual piece will fit. In that task users have to generate mental imagery of the puzzled image they have to construct. The figure below shows a sequence of the fish puzzle game surfaces.

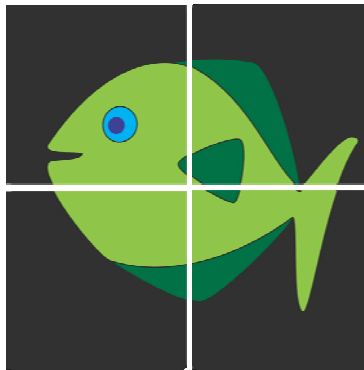


Figure 34. The Fish Puzzle game.

Scrabble game

The Scrabble game is essentially a word construction task. In the board game Scrabble, players form words by arranging tiles with letters printed on them. The tiles are shuffled around and the player tries to find associations. Based on these associations people set up the tiles to make it easier to form words.

In the scrabble game, the child has to form words out of letters by physically moving the tiles. Each tile has a letter on the surface and the memory requirement for the game depends on the number of tiles and on which letters the tiles have. The valid words for a particular tile-letter configuration should be uploaded to the tiles. The letter configuration should therefore be uploaded along with the game before the training activity. An example of word construction game is detailed in the Envisioning scenarios in the next paragraph.

The different game types above all have different game rules. These rules define the base of a game. Apart from them, different behaviour can be configured to support the game rules and the activity. This can for example be different output to the end-user to aid in accomplishing the task, i.e. the game. Other descriptions of the games with Active Surfaces are detailed in the Prototype Scenarios, see Par. 3.5.


3.5 Envisioning Scenarios

Two envisioning scenarios have been described in this paragraph. These are part of the tools that served to support the dialogue between the use and the architecture development. The Envisioning scenarios are a significant case of such an exchange. They served to describe Active Surfaces from both the use and technical perspectives.

The first scenario illustrates the Jigsaw Puzzle game presented above. It consists of an illustrated storyboard describing the use practices and the UML diagrams showing the functioning of the enabling architecture components. This envisioning scenario is the outcome of an interdisciplinary session involving the design team, the architecture developers from the Computer Science Dpt., University of Aarhus and the industrial designers from the Aarhus School of Architecture. The storyboard includes the scenes that define the situation of use, the corresponding technical UML diagrams, and a brief written description of what happens in each particular scene.

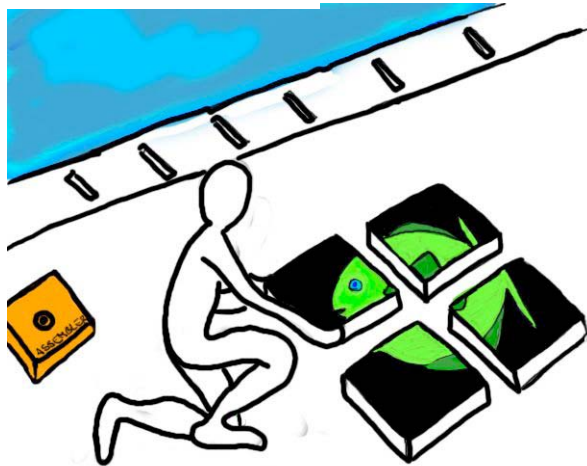
Envisioning Scenario 1

Scene 1

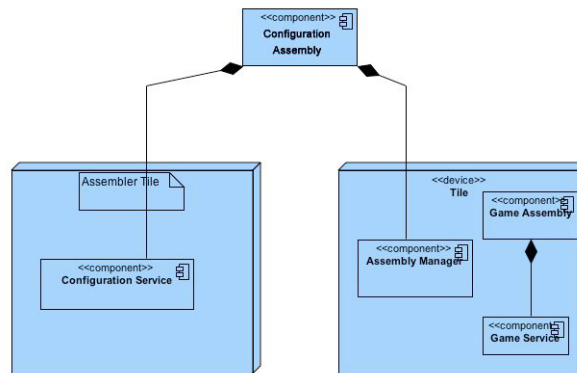


The configuration of the activity is performed outside the pool, e.g. at home or at the rehabilitation centre. There isn't a specific need to be in the vicinity of the pool from the system perspective. The therapist brings the assembler tile, attach it to the service browser and configure the activity by setting game parameters like the type of game, e.g. position game, and the output mode, e.g. using blinking lights.

Scene 2



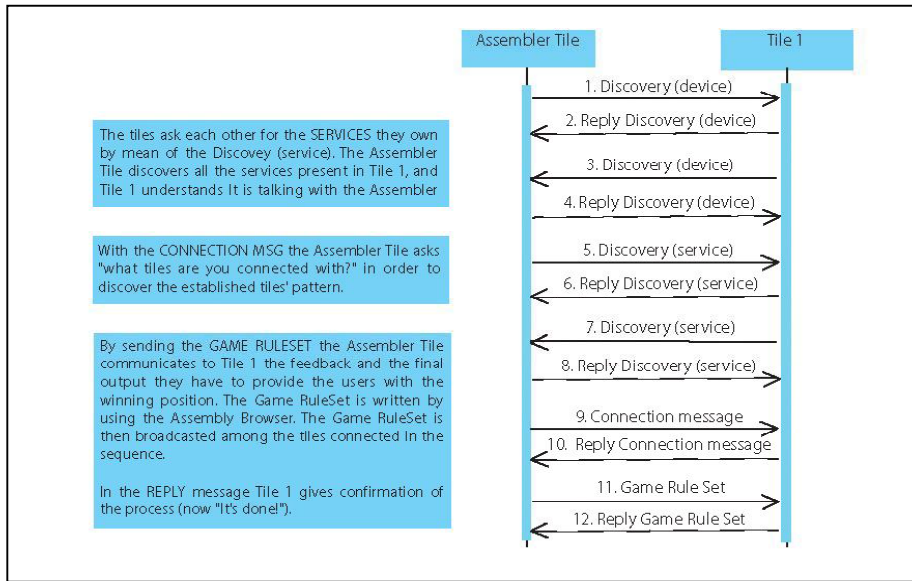
The therapist can now bring the Assembler Tile to the pool and align the tiles following the game she has previously designed. She positioned the tiles she wants to include in the game in the “winning” pattern.



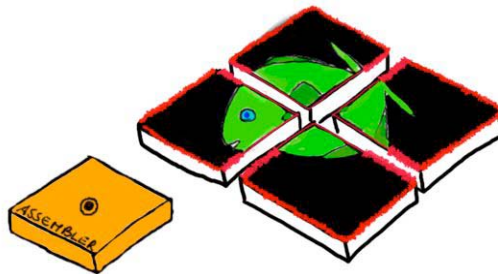
Scene 3



Now the Assembler Tile is wirelessly connected to the tiles and initiates a number of activities. By pressing the programming button the therapist sends a broadcast message to the connected tiles to remember their own and their neighbour's position.

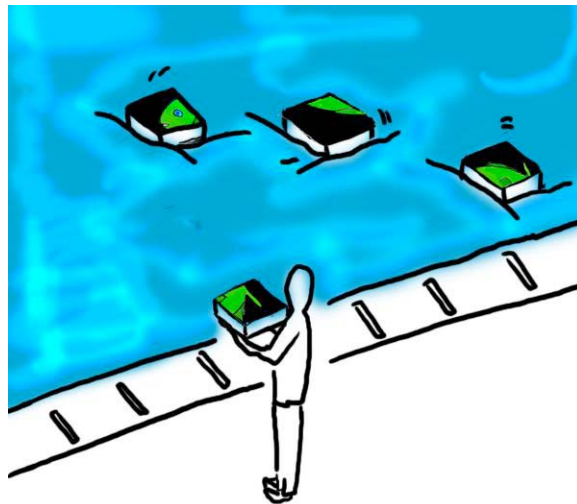


Scene 4

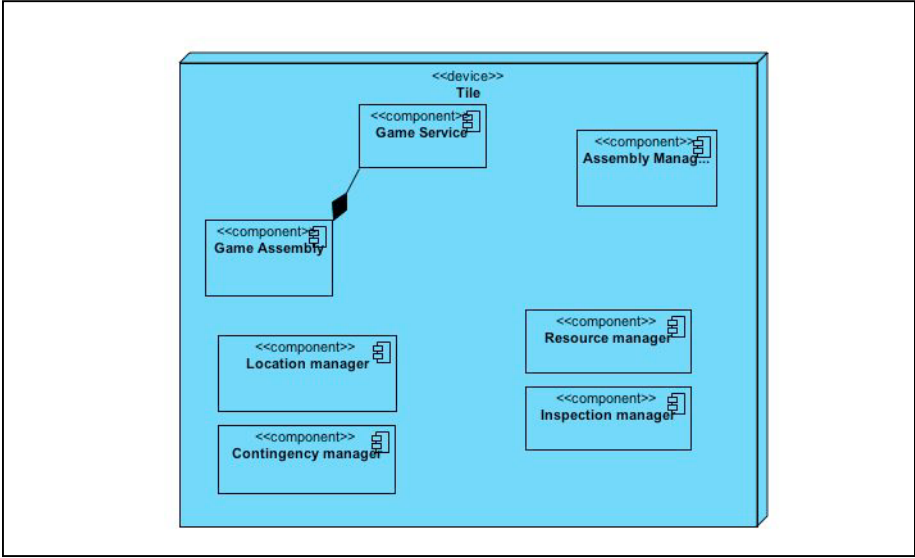


After executing this command the tiles have memorised their own and their neighbours' position and orientation. They also notify the therapist that the configuration of the Game Assembly is successfully completed through the AllHappy feedback.

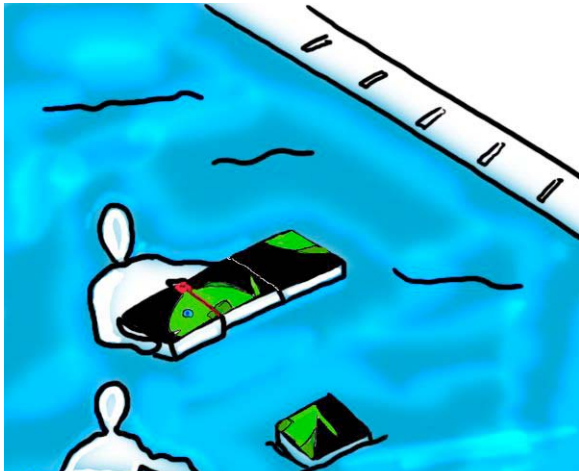
Scene 5



The therapist is now ready to start the activity with the patient and place the tiles onto the water.



Scene 6



The child tries different alternative sequences by moving the tiles around the water. When she puts two tiles correctly aligned these give a local feedback (SideHappy) whilst the final feedback is not given (AllHappy).

DEVICE DISCOVERY
Both the tiles continuously look for someone they can talk with. By receiving the REPLY they find that there is a tile they can talk with.

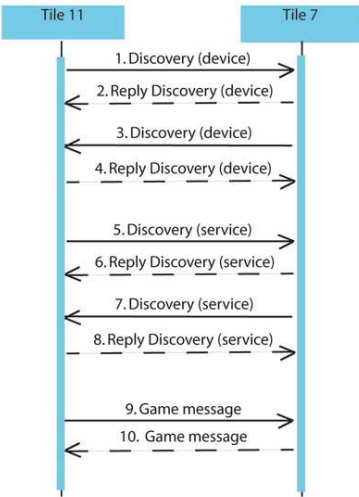
SERVICE DISCOVERY
Once having found a device (a tile), they both ask: "What service do you have? What can we do together?". By the Discovery (service) they inspect each other services.

If both the Tiles have the GAME SERVICE, they can continue to communicate. They find they have both the Game Service. The Game Service owns and manages the Game MSG

Tile 11 and Tile 7 exchange their own Game Messages. Each Game MSGs specify the Tile ID and the SIDE connections.

The Game MSG can be considered as the heartbeat the tiles continuously transmit at the current stage.

Since Tile 11 finds the correct tile (Tile 7) on the RIGHT side It changes its current status in SIDE HAPPY.



Scene 7



The child finally aligns all the tiles in the right position. This gives the final winning feedback (AllHappy). The game is solved.

The second Envisioning scenario describes the Scrabble Game presented in the previous paragraph in great detail. It is in a narrative form and tries to detail the game logics that underpin the Active Surfaces (Gronvall et al. 2006b).

Envisioning Scenario 2

The therapist configures the activity before the game takes place. Each tile is aware of its own unique ID, a dynamic label, and can propagate this data through its 6 sides. The tiles can also listen to incoming traffic on their 6 sides once they recognize that there is another tile present on a particular side via the device discovery and service discovery.

The therapist attaches the Assembler Tile (AT) and the Assembly Browser to one another. The PalCom device discovery now allows the Browser and the Assembler Tile to dynamically connect and exchange data. Once the devices have been identified, behaviours and game rules can pass from the Browser to the AT. The therapist configures the activity by setting those parameters in the tiles that constitute a game or therapeutic activity, such as the kind of activity (e.g. Position game or Scrabbles game) and the output mode (e.g. Blinking light).

In an initial version of Active Surfaces the AT will have a cable connection (i.e. RJ-45 Ethernet) between the system hosting the Browser and the AT. This connection will later on be wireless.

The therapist can now bring the AT to the pool and align the tiles she would like to include in the game, physically showing the tiles the 'winning' position and pattern. This activity will initiate device and service discovery activities inside the tiles. A tile now knows which tiles are connected to its sides and their orientation. The AT is now connected to the sequence of tiles. The therapist can rely on a simple physical interface on the AT: by using 'one touch' input they settle the tiles in the sequence. This command sends a broadcast message to the connected tiles to set their neighbour's ID and positions. This means that each tile looks at which other tiles are currently connected to its sides and saves this information together with the orientation of their neighbour tiles. The broadcast message can appear as follows:

AT to all Tiles: Set Neighbours(){ Set Identity & Orientation }

Now all the tiles have memorized their own and their neighbour's ID, position and orientation. The feedback the tiles have to produce when reaching the different Happy-states is now being downloaded from the AT to all the tiles. The tiles now notify the therapist when the configuration is successfully achieved by providing the settled output. They are now in their winning-position i.e. they are all in a *LocalHappiness*, which gives an overall *AllHappiness* state, which is possible because no *UnHappy* messages are being broadcasted.

At this point the AT asks for the current assembly of tiles to store the current assembly for later use as a record of the best practice, which serves as an inspiration for other therapists or patients activities.

The therapist is now ready to start the activity with the patient and throws the tiles into the pool. The child now starts to play with the Active Surfaces. The initial positions can be viewed in the following figure.



Figure 35. Initial configuration, the tiles are thrown into the pool. There is NoHappiness.

A *Discovery report* takes place as soon as the game starts. Each tile begins to report which tile is close to its sides and constantly matches these data with the stored *Neighbours* data.

In the early tentative trails the child tries different mistaken alternatives by moving around the tiles, which provokes new *Discovery* actions. When a tile is recognized on a side, a *getNeighbour* action is performed (on a service level). This gives the tile information about the Identity and Orientation of the tiles surrounding it. The message the different tiles broadcast is: *AllUnHappy*, because they are all in a not *LocallyHappy* state.

Step by step the child explores local solutions for the tile game, but has not yet found the global solution that solves the complete sequence. The child aligns two tiles that follow the right configuration, but the complete solution is still not found. This gives a local feedback that the two tiles are correctly placed while the final feedback is still not given as shown in figure 4.



Figure 35. HappySide for 'S' and 'H' tiles (Green light feedback).

Get Neighbors() (*S, H, I, F*) indicates *HappySide = true* for *S.East* and *H.West*. This indicates that *S* and *H* enter a *HappySide* mode on these sides.

The connected sides' lights on *S* and *H* are now turned on and (green light) feedback is given to demonstrate that the *HappySides* condition on these sides has been reached. For both the tiles the *LocalHappiness* state has not yet been achieved because it doesn't meet all of the required conditions for their happiness state.



Figure 36. 'F' is LocallyHappy (Red light feedback) and 'I' has one HappySide (Green light feedback).

Figure 5 shows another attempt to achieve the game solution. Given that the winning solution is the word sequence "FISH", F and I are in their right position and orientation. F now reaches a complete *HappySide* state and thus is *LocallyHappy*. This *LocallyHappy* state provides the user with a (red) light output depending on the initial configuration.

Get Neighbors() (F, I, H, S) gives *HappySide = true* for *F.East* and *I.West*. These condition makes *F=LocallyHappy* and *I=SideHappy*.

The global state of the active surfaces is maintained by broadcasting the message *UnHappy()* from those tiles that are not *LocallyHappy*. That puts all the tiles in an *AllUnhappy* state.

Going toward the final (and global) solution of the game, the child starts to align all the tiles in the right position by following the linear orientation of the word sequence.

Moving all of the tiles correctly produces the final output and the game is thus solved.

getNeighbors(F, I, S, H) gives *HappySide = true* for *F.East-I.West* and *I.East - S.West* and *S.East - H.West*. This results in all of the tiles (F, I, S and H) reaching the *LocallyHappy* state.

When all tiles are in the *LocallyHappy* state (i.e. no *UnHappy* message is broadcasted) the tiles reach the *AllHappiness* state. The game is solved as presented below.



Figure 37. All tiles are LocallyHappy, this gives a complete happiness within the system (AllHappy).

Chapter 4

Prototyping Active Surfaces

In this chapter the prototyping process of Active Surfaces is addressed and described in great detail. The features of the concept presented in Chapter 3 have been further developed into the PalCom architectural framework. We will describe the Qualities as they are realized and interpreted in the Active Surfaces application prototype (Par. 4.1). We will give a picture of the PalCom open architecture (Par. 4.2) specifying the components adopted in this application. Then the development of the Active Surfaces lightweight prototypes (Par. 4.3.1) and architectural prototypes (Par. 4.3.2) will be detailed. The outcomes of the prototypes development will be summarized and further elaborated in the Prototype Scenarios (Par. 4.4).

4.1 Architectural Qualities in Active Surfaces

The Active Surfaces concept and its implementation directly impact the meaning of the key Qualities that have been specified accordingly.

The concept emphasizes issues related both to the use, such as physical manipulation, positioning and emergent uses of the system, and the architectural platform, like the networking and dynamic assembly of tiles that is configured purposely (Pollini, Grönvall 2006; Grönvall et al. 2006). The concept also provides the possibility of exploring some of the indicative attributes for software quality, including understanding, ease of use, adaptability and error tolerance among the software ergonomic criteria (see the 'Ergonomic requirements for office work with visual display terminals (VDTs)' ISO 9241); and in addition scalability, reliability and reusability among the non-functional qualities, which can be used to judge the global operation of a system, rather than specific behaviours. Some of these are described in this section, as they are also the criteria that have guided the experimental plan described in this Chapter. In what follows we take these issues into consideration together with the Architectural Qualities introduced in Chapter 1, Assemblability, Adaptability, Resource awareness, Experimentability plus Scalability, and we try to see how they are embedded in and detailed by the Active Surfaces concept.

In contrast to the other PalCom application prototypes that specifically assemble heterogeneous devices (e.g. the assemblies in the Major Incidents-Overview scenario and the Neonatal Intensive Care scenario (Grönvall et al. 2007), in Active Surfaces entirely homogeneous devices, the tiles - which have exactly the same physical characteristics and computation and communication resources - are assembled. Each tile is an independent, physical, tangible object that can be picked up and moved around, and the interaction between the tiles is coherent and straightforward: all the tiles can communicate with their four adjacent neighbours. It implies that the system of tiles has structured connectivity, meaning a consistent pattern of connections between the components in the system that is based on physical proximity, i.e. each tile is accessible and the relationships are configurable. This is the basic organizing principle that characterizes the development of the tiles.

In Active Surfaces, complex patterns can be specified by simple component behaviours. In fact, each application running in Active Surfaces is supposed to be a small set of simple, interconnected or parallel services. Limitations on size and computation, along with the structural properties of the programming environment, encourage this simplicity. But the collective behaviours emerging across a set of tiles can be extremely complex.

The emphasis on simplicity also results in a network architecture characterized by homogeneity and enforced by local interactions. Each Active Surfaces tile is identical and interchangeable and can run any piece of code that is passed to it through a neighbour, included the game logics. They can be

assembled in many different formations that take into account the tiles' communication capabilities and the surfaces on which they have to be placed (Assemblability).

Each formation of tiles is instantiated as a functional and physical Assembly of devices and services. The Assembly, which is described in the following paragraph, takes form as the users construct it by means of the Assembler Tile. The Assembly can then be dynamically altered and adapted over time. Despite the stability it has when it is created, the Assemblies can be easily deconstructed and re-constructed in a different formation. This can occur by way of a tile substitution within the assembly or the instantiation of a new game activity. The tile assembly relies on the creation of connections among single devices that are supported by flexible ad-hoc networks that can be controlled and configured by end users.

Users may experience Active Surfaces as an assembly realized at two different levels. The basic assembly is created among the services present in each tile, such as the GameService, the LEDService or the TouchService. The individual assemblies instantiated in each tile are then programmed in an upper level assembly of tiles through which the game takes place. This aggregate of functions is then created in *assembly of assemblies*. The temporary or permanent assemblies can be created, deployed, maintained and deconstructed throughout time.

Due to the highly dynamic nature of the rehabilitation practice, the assemblies must also be dynamic, spanning over multiple devices and supporting the creative use and adaptation to emerging user requirements.

The embedded microprocessors adopted to build the flexible networks of tiles are at the same time powerful and resource constrained. In fact, the tiles are embedded systems that have all the hardware elements they need to operate (processor/microcontroller, ROM/RAM, I/O peripheral devices) and have limited available resources such as available energy, available memory or communication bandwidth in an embedded system (Resource Awareness). Because of the limitations of these devices they represent a concrete challenge for the developers of the software architecture.

The Active Surfaces system consists of a set of tiny, resource constrained computers that can be arranged together to create a physical network. Because the tiles can only communicate with their close neighbours, there is an explicit and consistent discovery and communication framework underpinning the whole system. The tiles can be arranged in three-dimensional patterns, like squares in a crossword puzzle, and tiles, which are stacked one on top of the other, communicate through the top and the bottom. Because of such communication rules, understanding the interactions between the Active Surfaces tiles is quite straightforward. The network can be easily reconfigured by picking up a tile and moving it; this movement immediately changes the feedback that is provided (Adaptability).

The key idea embodied in the tiles system is that a game application can exist within a network, rather than on a single unit or a central mainframe. Through the networking among the tiles and the instantiation of the assembly, they can discover the resources present in the system and debug the behaviour of such resources in order to overlook malfunctions or degraded individual or generalized performance. The resources are monitored and managed throughout time.

Active Surfaces can be thought of as a toy problem to experiment the PalCom Architecture because of its peculiar characteristics, as a modular system made of small easy to handle units. The tiles can be experimented with and tested without altering the structure of the system or causing any malfunctions or error that may potentially evolve into a failure for the system (Experimentability). Indeed, Active Surfaces has to operate even despite the presence of an error in the use. An error is a condition of

exception resulting from some deviation from the expected behaviour, which leads to a fault or failure, and the design of the architecture aims at minimizing the eventual adverse consequences of accidental or unintended actions.

4.1.1 Aspects of Scalability

The PalCom architecture is designed to be intrinsically scalable in order to cope with the complexity encountered in highly distributed, ubiquitous systems such as those enabled by the architecture. In the different application prototypes scalability primarily translates into the ability to efficiently manage the distribution, availability and configuration of resources according to the prevailing context and environmental conditions. Scalability is one of the major issues to have Active Surfaces as an application prototype for palpable technology since it indicates its ability to either handle growing amounts of work or to be readily enlarged (Bondi 2000).

Active Surfaces is designed both to scale vertically, scaling the PalCom architecture or part of it down to small microprocessors, and horizontally, scaling outward by adding more tile modules to the distributed system.

In fact, as for its specific conception, the number of tiles in a given configuration of Active Surfaces does not need to be fixed, since the mobile objects take the form of independent and distributed devices which can be added or removed on purpose. This realization of horizontal scalability also has the effect of reducing the system's tendency to break; rather, they gracefully degrade as single units or components of units, and thus the behaviour of the whole system is not affected.

Vertical scaling is particularly relevant to this research because experimenting with the PalCom architecture in the Active Surfaces scenario means exploring what happens when resources are removed from the single tiles in a system, typically CPUs or memory to a single microprocessor. This can be a scenario of progressive degradation of the system and its opportunity to face the loss of resources. Throughout the design and development process it has been debated whether to evaluate the performance of the architecture, or part of it, by experimenting with low resources microprocessors, or to use the full PalCom potentials in PDA-like devices with less constraints regarding processing power and hardware requirements. The choice was between the FS Forth-Systeme's UNC20 Universal Network Controller and more powerful processors (UNC90) with higher performance thresholds. The rationale for choosing the UNC20 microprocessors lay within their ability to challenge the architecture with hardware constraints, i.e. microprocessors and IR bandwidth. Another relevant constraint with the development of Active Surfaces is power consumption; how well the different hardware implementations perform with respect to different battery capacities has to be tested.

The use of tiny platforms proved to be relevant to Ubiquitous Computing applications and began to change the way we think about distributed and mobile computing in general. In order to develop small, simple and physically manipulable devices the PalCom middleware-networking infrastructure has to be implemented and experimented with respect to the Qualities presented in Chapter 1.

In particular in our experiments we take into account more operative specifications of the qualities described above, such as real-time constraints, performance throughout communication and discovery and support for re-configuration.

The software components running on the Active Surfaces are subject to real-time constraints due to the interactive and embedded nature of the tiles. Like many embedded systems, they engage in an elevated interaction with the outside world through sensors, actuators or wireless communications. Such interactions result in more or less strict timing requirements depending on the objectives of the application. When the system behaves appropriately it depends on the produced results, the system's *functional* correctness, but also on the time when they are produced, the system *temporal* correctness (Bass et al. 2003).

In this section we outline related work within middleware for ubiquitous and pervasive applications. The following projects adopt a large number of tiny computers in several applications and emphasize the distributed, embedded and pervasive nature of UbiComp. They are especially valuable with respect to scalability and in relation to the software architecture through which they are realised. We survey the architectures scaled down to resources constrained devices, especially those lower than PDA (i.e. mobile phones and microprocessors). The research projects have been outlined in the table below. The three research projects highlighted in the table are among the very few cases in which middleware architectures have been scaled to resource-constrained devices.

Architecture	Ref.	Scalability
> Orbix/E Lightweight CORBA ORB	IONA Orbix/E Datasheet	CORBA optimized for embedded applications. Orbix/E has a relatively small memory footprint and provides the ability to choose a subset of features for a given application in order to optimize its size and speed.
> Prism-MW	Malek et al. 2006; Mikic-Rakic, Medvidovic 2002	Distributed systems applications running on resource-constrained devices, with low amounts of memory (e.g., 256 KB on the Palm Pilot) and slow processing speed. Running on Motorola 68328 processors at 16 Mhz, with 512 KB (Personal) or 1024 KB (Professional) built in memory.
> OSGi – Concierge	Rellermeyer, Alonso 2007	Fluid Computing; flowSGi running on Intel iMote2 sensor network board (ARM5-TE compliant Xscale PXA27x, 32 MB RAM, 32 MB Flash) and Linux OS for arm-embedded and a JavaVM installed.
QuAMobile	Amundsen, Eliassen 2006	Quality of Service middleware for mobile computing running on Mobile Phones.
HP Cooltown – Coolbase	Kindberg, Barton 2001	Web-based client-server architecture involving handheld PDAs, smart printers, mobile phones, scanners.
Jini	Waldo 1999	Mobile phones, PDAs.
xMIDDLE	Mascolo et al. 2002	Data-sharing middleware for mobile computing. XMIDDLE is lightweight and fast, and caters to the frequent disconnection behavior that mobile devices exhibit. Running on PDAs, Mobile Phones.
Objc/ Speakeasy	Edwards et al., 2002a, 2002b, 2005; Newman 2002	Networking and connection infrastructure. A desktop computer can provide mobile code execution services for other attached devices (such as speakers, digital cameras, printers, scanners). Running on PDAs.

Table 38. Related project and scalability

The vision of making Ubiquitous Computing palpable, described in Chapter 1, ought to be purposely realized. In this research we consider scalability as major requirement for the Active Surfaces scenario. Surveying the middleware architectures for Ambient and Ubiquitous Computing, we observed that only a few of those that have been put on the table may be scaled down vertically and be embedded in resource-constrained microprocessors.

Indeed, most of them require high computational resources and only the Orbix/E Lightweight version of the CORBA ORB platform, the Prism-MW (Programming in the Small-and-Many Middleware) and the Concierge release of OSGi have also been experimented with microprocessors smaller than common PDAs and Palmtop.

All the outlined projects deal with scalable platform and PDA-like powerful mobile devices but very few can rely only on resource-constrained devices, with the characteristics described in the above paragraph.

In the next paragraph we will describe the infrastructure platform that realizes the vision of palpable computing and that may fit with the needs of the Active Surfaces scenario by being scaled down to very small microprocessors (e.g. Atmel UNC20 serie).

Together with the experiments with embedded systems we have also been concerned with the simulation of the applications developed by using the PalCom architecture. In particular this has been done to create a simulation of networked devices running on a single desktop machine.

The comparison between embedded tiles and simulated tiles also reveals different criteria with regard to power constraints, 4 or 6 sides' communication, speed of execution, robustness, etc.

Simulation is a powerful technique because the developer can exert fine-grained control over the simulated world. It is possible in a simulation, for example, to carefully balance the complexity of the system against understandability. Furthermore, in a simulation the problems of performance and speed of execution are skipped in favour of the possibility to vary and repeat the task in a homogeneous environment. The advantages of simulation are the chance it provides to create a dedicated environment from scratch, tailored to encourage certain kinds of learning and exploration. The Active Surfaces architectural simulated prototype described in the future paragraphs will provide a valuable example of the advantages of simulation in the domain of Ubiquitous Computing software architecture.

Certainly we also have to consider that a simulation is limited in showing the effect of the real-time interaction with the world. Working in simulation suggests a particular relationship between user and system, which is different from those that exist in systems that users perceive as real, working and embedded.

The Active Surfaces palpable 'system' ought to exhibit these qualities in a way that is appropriate to the application context. The manner in which these qualities are reified and implemented is the choice of the architecture developer and the designer, respectively. It can be done through both design and architectural choices. The PalCom open architecture to develop the key Architectural Qualities is described in the next paragraphs.

4.2 The PalCom Open Architecture

The palpable Active Surfaces system that we have designed for the therapeutic practice in the water relies upon features that are coupled with the concepts and components presented in this section. They depend upon the reference implementation of the palpable software architecture, as established by the

PalCom project. As detailed in Par. 1.3, the software architecture was developed together with the theoretical framework. The infrastructure that we present in this paragraph has been developed to reify the Qualities for palpability addressed by this research. In what follows the Open Architecture has been outlined; the prospectus is based on the PalCom Deliverable 54. The features marked with (*) are those that are specific to the PalCom architecture.

Service Oriented Architecture

- Loosely coupled independent communicating services on network-enabled devices
- PalCom Services instantiated from components
- Services can be stateless or stateful and are visible on the network
- Compositions of services into assemblies, including non-PalCom services *

Platform-Independent Runtime Environment

- PalCom Virtual Machine (Pal-VM) with multi-lingual support
- Pal-VM exposes explicit low-level resource consumption enabling resource awareness *
- Loads platform independent components compiled from Java and/or Smalltalk
- Hosts (executes) services instantiated from components
- Communication primitives

Communication

- Open communication protocols
 - Human-readable representation, yet highly compact and performant
 - Pluggable Media Manager adaptors to any protocol
- Adaptive connector selection according to situation and needs of user *
- Support for publish/subscribe (multicast), point-to-point messaging (one-shot), point-to-point connections (e.g., streaming, client/server)
- Transparent support for multiple bearers (Bluetooth, Ethernet, Infrared, etc.) *

Lightweight Resource Management Middleware

- Assemblies are the applications *
 - Glued together from high-level resources (services/ devices/ actors/ connectors)
 - Open specification of human-readable assembly descriptors (XML)
 - Scripting to create active compositions of services, devices, etc.
 - Maintains mandatory/optional, loosely/strongly specified bindings
 - Constructed using visual browser or programmatically
- Resource Management
 - Persistent discovery of services/devices
 - Provides resource descriptors for Assembly Manager
 - Awareness of, and adaptivity based on low-level resources *
- Contingency Management
 - Maintenance of assemblies in cooperation with Assembly Manager
 - Reactive and proactive problem compensation possible *

Open data representation on network nodes

- Hierarchical Graph datastructure (Hgraphs) for inspectability *
- Uniform access to all data
- Remotely accessible
- Unlocks Object-Orientation encapsulation
- Reflection and Inspection *

The Palpable Computing architecture is a middleware Service-Oriented Architecture (SOA) (Brønsted et al. 2007) developed as a set of small self-contained units that solve user tasks through intercommunication. In Service-Oriented Architectures a number of services can communicate among each other in order to provide support for both end-user applications and for other services through published and discoverable interfaces. The services are autonomous and can be located on different platforms in a distributed network.

The PalCom architecture provides that these distributed services are connected through the use of Assemblies. The Assembly is the core concept of the architecture and three middleware managers exist to support the formation, the modification and whole lifecycle of the assembly itself, the services and other components. These are the Assembly Manager, responsible for the creation and lifecycle management of PalCom Assemblies, the Resource Manager, which maintains an up-to-date directory of all active (and if required, inactive) software resources, and the Contingency Manager, which maintains resource and assembly resilience by applying both reactive and proactive compensation policies and mechanisms (PalCom Deliverable 54). They are represented in the figure below within the whole computational infrastructure that realises the concepts of Palpable Computing.

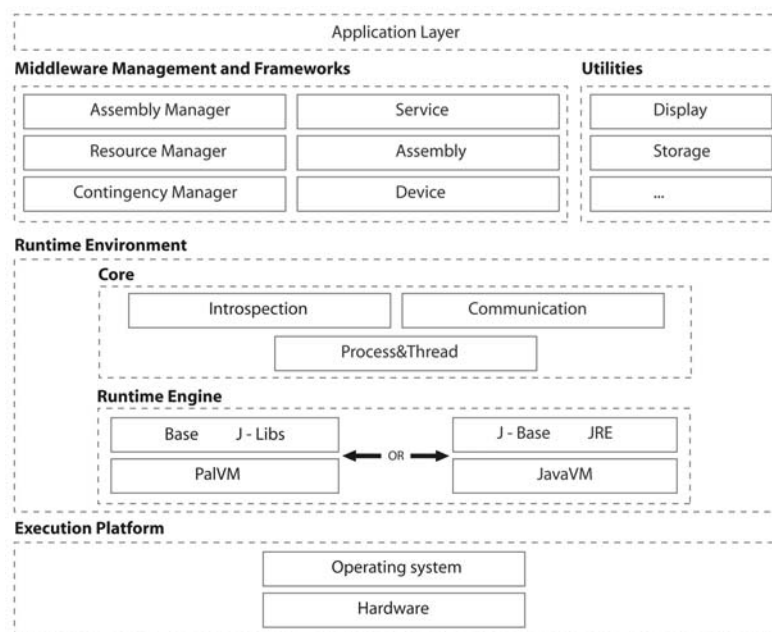


Figure 39. Layers and Functional Elements in the PalCom Infrastructure (PalCom Deliverable 54).

The boxes in the figure represent functional blocks of code at runtime (PalCom Deliverable 54). Elements in one layer are dependent on the lower layers. PalCom assemblies and services build on the Application Layer with support provided by the lower layers.

The PalCom architecture is *open* in the sense that it allows for the open-ended composition and development of PalCom components and that it allows for interoperation with non-PalCom services. In fact, the architecture is released as an open source with the full source code under the BSD license. Users can develop applications starting from a PalCom architectural toolbox that can be downloaded in a binary, ready to use version of the architecture. Developers can easily enter the PalCom source code and develop applications for Ambient and Ubiquitous Computing environment. The architecture is *open* also because it allows the potential inclusion in each device of a readable XML Descriptor in

the PalCom Assembly. Application scenarios involving the integration of non-palcom devices with palpable computing have been experimented with throughout the project.

The architecture is detailed in the following sections with a focus on the components utilized in the Active Surfaces application prototype. The different kinds of Resources, the Assembly, the Services and the Devices as basic concepts of the PalCom architecture have been described to a great extent together with the Middleware Managers. In addition, we have also described the Runtime Environment (RTE) and the PalCom Virtual Machine (PalVM), as the Runtime Engine is for the low-level structuring of programs used in the Active Surfaces. We also present the Storage component, a communication-specific utility contained in the utility module, and the characteristics of the execution platform, the underlying device hardware on which palpable computation is hosted. The descriptions below are based on the PalCom Deliverable 54.

4.2.1 Main Concepts

- **1st Order Resources.** 1st Order Resources are low-level resources associated with a physical device, including processor load, memory, bandwidth and power supply. Since Palpable Computing considers applications made of distributed populations of services, the consumable hardware resources and their management play a fundamental role in the palpable system. The deployment of a service (i.e. a 2nd order resource) depends on the availability of the foreseen hardware requirements. The Resource Manager is deputed to free up resources through priority overrides, scheduling, etc, within the operational constraints of service, devices and usage expectations.
- **2nd Order Resources.** 2nd Order Resources are abstractions used to describe those resources that either contain or consume 1st Order Resources. Examples include Assemblies, Services, Devices and communication channels. In order for the Palpable Architecture to be flexible and robust enough to support the user's experience of palpability, the 2nd order resources ought to be directly manipulated and adapted.
- **Devices.** A Device is a software and hardware unit that hosts a Runtime Environment and within it, components, Services or Assemblies, which are deployed onto Devices. The Device framework provides points for initialization and termination of services and managers on a device (PalCom Dev 54). Each Device has a unique identifier, the DeviceID. The DeviceID identifies the device both for incoming messages, in which case the URL of the sending device is translated into the DeviceID, and for outgoing messages, in which case the Media Abstraction Layer translates the DeviceID to the matching URL. The interface Device IO is used to communicate with the hardware on a device by means of low level processes such as interrupt routines. As a support for the developer, the PalCom architecture also provides a framework for simulated devices. Simulated devices are Java programs that run on a desktop machine, with graphical representations of the hardware of a Device. Code written using the device framework can be run in simulated devices first, and be evaluated before being deployed onto real devices. That offers advantages in terms of easier debugging and the easy creation of multiple devices for testing purposes. It also allows the envisioning and experimentation with future Ubiquitous Computing applications in the absence of embedded prototypes.
- **Services.** Services are discoverable computation units that are available to the users of palpable applications. They are the primary constituents of Assemblies within the Runtime

Environment. A Service encapsulates a remotely accessible, discoverable, self-contained, reactive, and possibly context-independent element of infrastructure functionality (PalCom Deliverable 54). Services are remotely accessible because they can be accessed via different forms of networked communication systems.

Services are reactive or proactive with respect to the received events that conform to its published interfaces. Reactive implies that a Service will perform some actions in response to the trigger of the service operation. It is entirely possible and expected that services may also operate proactively and hence, autonomously. Being reactive does not preclude the ability for a service to be proactive and a Service can also be a combination of classic reactive and proactive methods. The context information needed for proper Services execution must be provided when initializing a Service, such as information concerning other services to be used (PalCom Deliverable 54). Services can augment this information during execution if the context changes.

Examples of services adopted in the Active Surfaces application prototype are the Communication Service, the Discovery Service, the Touch Service and the Feedback Services (e.g. LED Service, Vibration Service).

- **Assembly.** A PalCom Assembly is a composition of PalCom 2nd Order Resources, potentially including other Assemblies. It consists primarily of Services and provides additional functionality for the end-users beyond that provided by the individual services (Svensson et al. 2007). PalCom Assemblies are relevant both for the developer of PalCom systems as a means to build new functionality, as well as for end-users to dynamically tailor and augment the behaviour of the system within a concrete context (PalCom Deliverable 54). The PalCom Assembly may interact directly with the users or act autonomously at the user-level application. The resources distributed over a number of devices are constructed in ad-hoc networks within a given operating vicinity. To be included in the assembly each computational resource has to provide an XML Descriptor with essential data regarding how to deal with it (functionalities, communication channels, features). The computational resources are discovered and composed accordingly by means of the interpretations of XML Descriptors. Any available resource in the operating environment that matches with the requirements declared in the Descriptors is identified and may be invoked to a particular form and assembly. They also specify the rules for the governance and replacement of Service. Essentially, Assembly Descriptors contain instructions for composition and for coordination. Composition is the ability to specify and plan a set of PalCom 2nd Order Resources that contribute to the behaviour of a PalCom Assembly. Coordination describes the interaction of those resources within the context of a PalCom Assembly. The model is dynamically constructed with the opportunity to break it apart and reconstitute the computational resources.

An Assembly may be distributed over multiple physical devices and, in this case, include the descriptors of communication channels required to connect the remote parts. In fact, some 2nd order resources must be stationed at the assembly that uses them, while other resources can be remotely positioned. The actual location of a 2nd order resource exploited in an assembly can be defined in the Descriptor of the Assembly. It defines the 2nd Order Resources present within the assembly, which will primarily be Services, and how these Services can be composed to offer aggregated behaviour. When an assembly is no longer needed, it can be decomposed only following the user initiative (Svensson et al. 2007). Decomposing an assembly simply involves unbinding the 2nd order resources that are bound together according to the Assembly Descriptor. The composition of an Assembly may be fixed for the duration of the Assembly

instantiation, or it may change dynamically, both in terms of the individual resources and the way in which they are composed (PalCom Deliverable 54). Once an assembly has been created it has to be maintained and it includes Deconstruction when the assemblies are no longer needed, and Re-construction when an assembly is subject to a partial failure that can possibly be resolved by dynamically replacing parts of the assembly with other (in the operating environment) available second order resources equivalent to those failing. Static assemblies will not vary in structure throughout their lifetime. Dynamic assemblies, however, can be dynamically altered at runtime to add or remove 2nd Order resources, if this is required by changes in operational requirements, constraints, context or contingency actions. Given the dynamic nature of the environment and the human activity, maintenance and control of the assembly is required. The maintenance of assemblies includes the re-construction when (even partial) failures occur. Because of the real, hydiosyncratic context, a failure can consist of resources appearing and disappearing and operational disruption. The assembly can be inspected by means of different strategies (manual and automated) in order to handle the failures and to recover from breakdowns.

4.2.2 Middleware Managers

The Middleware Managers are shown on the right side of the diagram, executing on the PalCom Runtime Environment and managing PalCom Assemblies and other 2nd Order Resources.

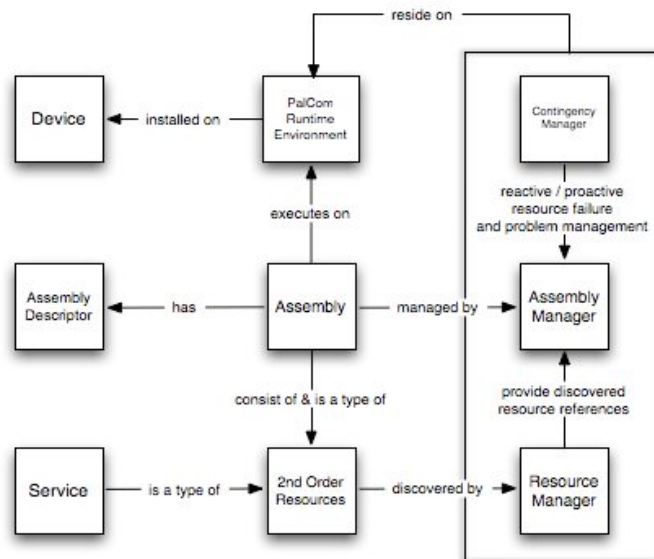


Figure 40. Middleware Management and its relationship to the concepts in the PalCom Architectural Ontology (PalCom Deliverable 54)

- **Assembly manager** The Assembly Manager primarily has to deal with the construction of Assemblies according to the underlying XML-encoded Assembly Description, which will contain details of the particular 2nd Order Resources and how they may be used. Both templates and instances of 2nd Order Resources, and specifically Services, can be specified in the Assembly Description. Together with the construction of Assemblies, an Assembly Manager is responsible for deploying the PalCom Assembly and thereafter managing its lifecycle in collaboration with the available Contingency Managers.

- **Contingency manager.** The Contingency Manager is responsible for indicating faults and problem conditions occurring in active PalCom Assemblies and 2nd Order Resources through the application of a variable set of contingency tools and mechanisms. Contingency implies the ability of a palpable system to automatically identify problem conditions, determine suitable means to resolve them and then apply appropriate mechanisms to prevent future error conditions. The Contingency Manager is expected to provide a set of reactive contingency actions (i.e., compensations) that respond to unattended events. This provides that a system becomes more resilient and also capable of adapting to changing ambient conditions. It aims at maintaining the Assembly's stability over time and as such Contingency management ensures that 2nd order resources in the Assembly are always available or, if not, how they might be replaced (see PalCom Deliverable 54 for further details). It also monitors the performance of 1st Order Resources on specified devices. If a threshold is crossed, a contingent action can attempt to trigger the re-balancing of resource load across additional devices. It can also establish proactive precautionary error avoidance strategies, which will attempt to mitigate potential error conditions.
- **Resource Manager.** The Resource Manager is responsible for maintaining an up-to-date directory of discoverable 2nd Order Resources. The primary operations of a Resource Manager are the discovery and monitoring of 2nd Order Resources, the match, the description and the provision of resources according to needs expressed by Assembly and Contingency Managers. The Resource Manager manages the lifecycle of selected resources including initiation, termination, restarting and movement, etc. 2nd Order Resources (PalCom Deliverable 54).

4.2.3 Runtime environment

The Runtime Environment is a platform for realising and deploying PalCom Components and provides an execution environment for Runtime Components, Services and Assemblies.

The PalCom Runtime Environment explicitly relies on the existence of a network and the capability of PalCom Devices to join and leave this network when necessary. It provides an infrastructure for inter-device communication that hosts and executes 2nd Order Resources and the Middleware Managers described above.

The Runtime Environment handles the hosting of Processes and Threads, Communication and Inspection Components. The Process & Thread model specifies how to create parallel or alternating sequences of actions within the RTE, and provides mechanisms for various levels of isolation between such units of functionality. The Communication model defines mechanisms that PalCom Services can use to access and interact with other PalCom Services residing on other PalCom Devices. At the lowest level of the communication model, the Media Abstraction Layer (MAL) serves as a bridge between the PalCom communication model and the actual network used. This layer mediates between the implementation details of the used network (Infrared, Bluetooth or IP) and the rest of the PalCom architecture. This is essential in order to make it easy to add support for new network technologies in the future, and also to make it transparent to dynamically switch between available network technologies. The communication is organized through the use of different protocols, the Discovery protocol and the Service Interaction Protocols. The Discovery is used by PalCom Devices to announce themselves and to present the capabilities they offer to other Devices and to end-users (PalCom Deliverable 54). In the case of the implementation of Services of a small device, only a part of the protocol must necessarily be implemented, namely the part that responds to heartbeats and requests.

The Service Interaction Protocol is used after services have been discovered using the PalCom

Discovery Protocol. It is used for establishing connections between the Services, also while they are operating. In the standard Service Interaction Protocol, communication takes place in an asynchronous manner, in the form of commands sent over connections.

4.2.3.1 The PalCom Virtual Machine – PalVM

The PalCom Virtual Machine (PalVM) is a dedicated virtual machine developed within the PalCom project. It has been developed as a consequence of the requirements coming from the PalCom challenges, the fieldwork analysis and the various application prototypes, including the Active Surfaces. The PalVM is a resource aware engine that may observe and manipulate 1st order resources such as memory, CPU load, battery stand and storage utilisation as well as second order resources such as processes, components and devices (PalCom Deliverable 24). The PalVM must provide the necessary support for fundamental contingency handling and means for controlling processes, threads, and services.

The Core Libraries, the Middleware Managers, and the Utilities are all written in Java code, compatible with pal-j, and they can be compiled with either the Java Compiler (javac) and executed on JVM, or with pal-j and executed on the PalVM (PalCom Deliverable 54). When the architecture runs on a desktop machine, the code can run on either the Java Virtual Machine (JVM) or the PalVM. Instead in small devices, like mobile phones and resource constrained embedded systems, the PalVM has to be exclusively adopted to implement the PalCom architecture. In fact, it is possible to implement the PalVM on standard operating systems and also directly on a microprocessor without requiring an operating system. Currently the PalVM can be used on Intel based Linux, ARM-7 and ARM-9 based Linux, Intel based Windows XP, Intel and PowerPC based Mac OS X 10.4.

The open architecture and palpable runtime is designed to also be portable to new hardware platforms and to support RAM (lower bound in the order of 256Kb) and slow, low-power microprocessors. The portability of the PalVM to small devices, i.e the ARM-7 microprocessors, is one of the core issues in this research study.

The challenges of heterogeneity and scalability require experimentation with running the PalVM on small devices like sensors, battery-powered sub-PDA or memory-constrained devices. Experiments with such small devices like those that are described in Chapter 5 require very different considerations with respect to those on medium sized PalCom devices (PDAs or stationary devices with large amounts of memory, for example 16Mbytes).

The goal for the PalVM is to be able to run on devices with only 256 kbytes of memory. When compiled for the ARM processor, the current implementation of PalVM takes around 122 kbytes. In order to run on embedded devices with Linux, the PalVM must be linked to the system libraries and this provides that the current size is 297 kbytes. This example is based on benchmark on the PalVM version running on the ARM-Linux based Axis cameras used in the Sitetrack application prototype (Buscher et al. 2007). The experiments presented in this thesis depend critically on such a performance.

In the Active Surfaces system, the tiles are small building blocks that are distributed in the environment and can be composed together in flexible ways in order to build larger systems. In order to realize this concept, the modules have to be small, both in size, cost and power requirements. A

simple way to save power is to run a system at a lower speed, so it is also important that the software run as fast as possible on a given hardware with a given CPU speed. In order to facilitate the development of prototype applications for PalCom it is also important to have an implementation that exhibits the qualities of speed and economy of space. After the initial proof-of-concept virtual machine, the PreVM, written in Java, the PalVM specification was implemented again and rewritten in C++. One of the main reasons for this move was to improve the execution speed, especially because the performance of the PalVM was still not satisfactory for those prototypes that were based on ARM processors, like the UNC20-based devices (i.e. the tiles).

4.2.4 Execution Platform

The Execution Platform consists of device hardware and optional operating system(s) running on that hardware. The execution platform is a prerequisite, but this architecture does not mandate any details regarding technology choices or specific configuration options, however, since PalCom is a middleware architecture, communication capabilities are required.

Many 1st Order Resources are directly related to the Execution Platform, e.g. memory, cpu cycles, communication bandwidth. The Runtime Engine provides primitives for accessing these 1st Order Resources, possibly through an Operating System (PalCom Deliverable 54).

4.3 Prototyping Active Surfaces

Active Surfaces gives the opportunity to test concepts and components of the Open Architecture through the use of an interactive and distributed system of tiles. The tiles are independent from a central system and are a valuable application prototype for testing and evaluating a service oriented architecture like the one developed in PalCom.

The use of prototypes in computer science has its roots in an acknowledgement of what the prototypes introduce in engineering. In engineering they are used toward the end of the process in order to represent the final system, while software prototypes are usually used much earlier in the process (Bardram et al. 2005). In fact, they are often not supposed to represent any final or complete stage, but rather a working and usable set of functionalities that can be used for user evaluation.

Throughout the project the application prototypes have been used to demonstrate a wide range of challenging features that are evaluated in respect to how usable the architecture is for users and/or developers. Together with the user testing, measurements of performance parameters are also required to evaluate software architectures. In fact, they allow certain user experiences by means of an enabling platform for the development of applications that must fit with the requirements of the application prototype.

Architectural and design aspects of the application such as those described above have been explored in several interactive prototypes that have been developed following iterative cycles of design sessions, user evaluation and architecture development. We adopted a step-wise approach to transform our concept into early mock-ups and then into working prototypes (PalCom Deliverable 18; PalCom Deliverable 50).

At the beginning of the prototyping we adopted a variety of techniques to allow for early experimentation, i.e. experimentation with devices exhibiting palpable use properties, before the architecture was properly implemented. This was realised by means of mock-ups and lightweight working prototypes. Mock-ups have been developed so far in order to allow for the earliest

experimentation and concretization as possible (PalCom Deliverable 18).

A number of low-fi prototypes have been developed to verify the different assumptions, the potential design solutions and the integration with the swimming pool setting. This work has resulted in more mature architectural prototypes that have been assessed by the stakeholders throughout the evaluation process. The users have been able to comment on functionality, look-and-feel, and have also been able to use ‘semi-working’ prototypes in a way that is not permitted by the traditional Wizard of Oz approach. In reality, because of the complex interactions occurring with the experiment with interactive distributed systems, some of the prototypes turned out to be easier to develop than setting up the Wizard of Oz sessions reported in Par. 3.3.

As a primary step for the development of Active Surfaces we have been focusing on embedded tiles floating on the surface of the water. The adoption of floating tiles seemed to be more promising for the exploration of the concept and for consolidating the functionalities through iterative prototyping and evaluation with end-users. The future development of the Active Surfaces application would also regard vertical configurations of tiles on the edges of the pool and carpet-like bottom surfaces to be used as tools for the therapy and the re-education of delayed abilities.

The early prototypes of the tiles have relatively limited input and output capacities; each had four PowerLEDs, one for each side, and the activity needed to be carefully tailored to work well with only moderate visual feedback. A common characteristic of the Active Surfaces prototypes is that they communicate using only a low bandwidth, short-range infrared link and have limited output capabilities, such as the set of LEDs available in the current implementation.

Throughout the prototyping and architecture development entire devices and their services have also been simulated in order to experiment with their use in a network of PalCom services without the constraints of the hardware implementation. In this case the embedded prototypes are usually advanced prototypes close to the final system, which might be tested in the appropriate implementations.

The first of three Active Surfaces prototypes we developed is the *Lightweight embedded prototype*. This is a lightweight, non-architectural prototype that expresses some of the core objectives of the infrastructure. It leverages the fullest extent of the power of the infrastructure, and how this power can make a difference (Edwards et al. 2003). It serves as proof-of-concept application that succeeds in demonstrating compelling new user experiences.

The other two prototypes are architectural prototypes that have been developed contemporarily. Bardram (2004) has defined architectural prototypes as consisting of a set of executables created to investigate Architectural Qualities. In the initial development phase the Qualities are related to concerns raised by stakeholders of the system under development. Architectural prototyping is the process of designing, building, and evaluating architectural prototypes. It is a viable and cost-efficient technique for exploring an architectural design space (Bardram et al. 2005) and for addressing issues regarding quality attributes and other non-functional qualities. The experimental and evolutionary architectural prototyping aims at facing a number of challenges and trying out new ideas. The two Active Surfaces architectural prototypes were the *Simulated Architectural prototype*, the *TilesSimulator*, and the *Embedded Architectural prototype*.

The TilesSimulator was built and operated as a GUI-based application until the hardware implementation could be fully reached. The simulated architectural prototype is used to determine whether a software solution that has already been implemented is adequate or not. It meets the specific requirements regarding execution speed and flexibility that is needed in the final system.

The embedded architectural prototype allows concrete measurements to be made under a range of different situations. These situations are defined in terms of Qualities attribute scenarios that identify major performance concerns.

These architectural prototypes are both experimental and evolutionary (Bardram et al. 2005), they have been gradually developed to evolve into the final system either in an incremental way or by iteratively refined cycles.

In the following sections the design and development of the three Active Surfaces prototypes have been described. The lightweight, non architectural prototype is described with much more detail as being mainly designed, developed and evaluated by the University of Siena while the other two architectural prototypes have been developed in collaboration with the University of Aarhus, Denmark and the University of Lund, Sweden, both partners in the PalCom project, with a major responsibility on their sides.

4.3.1 Lightweight embedded prototype

There have been two versions of the Lightweight embedded prototype. The initial prototype was developed as an exploratory prototype used as a means for gathering user requirements. This prototype had limited functionalities, but it worked sufficiently in as much as it could be more easily adapted to new user requirements as they emerged. The current Lightweight prototype is instead an evolution of the initial version both from the hardware and software perspectives. Nonetheless, because they are different, the two versions of the Lightweight prototype have both been conceived as horizontal prototypes. They focus on what is the role, the look and the feel (cfr. the prototype model, Houde, Hill 1997), and the underlying functional architecture is often hacked or implemented with alternative architectures.

The creation of lightweight applications purely demonstrates the utility of the novel aspects of the infrastructure. In contrast with the engineered applications of the architecture, these applications are easy to build and require few engineering resources. Lightweight applications can be used to evaluate the system in the actual context of use. Such an evaluation requires that the system be robust enough for permanent use, support features required by the context - such as being waterproof in the swimming pool scenario - and provide support for migration to the architecture (Edwards et al. 2003).

The use of Active Surfaces lightweight prototypes provided for the early and advanced testing of core infrastructure features without developing the architectural applications. In this way it has been possible to get the basics of the infrastructure tested rather than solely an exclusive focus on the functioning applications.

The method adopted in the early and advanced testing with the Lightweight prototypes was an empirical observation of the emergent uses of the prototype and the interactions occurring. The empirical observation of the real use of the non-architectural tile prototypes has been essential in proving the Active Surfaces application's significance. This evaluation still does not probe the

realistic use of the final architectural system; rather it provides pictures of which user experiences it might support and guides the design.

Both the initial and the current Lightweight embedded prototypes were developed as a result of iterative design cycles. Building a flexible system from both the perspective of both the hardware and the software was a primary goal for the lightweight prototypes.

4.3.1.1 Initial Lightweight embedded prototype

There were four iterative prototyping cycles to develop the initial Lightweight embedded prototype. It was made of three tiles, which communicated with each other on two sides, thus creating a unique, hard coded sequence of tiles with fixed positions. Each tile was a tiny computer, with a Basic Stamp 2 microcontroller serving as its functional core. In the tiles system, computational power resides essentially in microcontroller speed, memory size and communications bandwidth. In addition, in order to put their computational abilities to meaningful use, the tiles must be able to receive input from and provide feedback to their users. Each tile runs on batteries and requires specific attention to support this power supply.

The first step in designing the tiles was to choose the Basic Stamp microprocessor. In a network environment, effective computational power is often limited by communication speed; a fast processor does you little good if that processor is always waiting to send or receive data through a narrow communication channel. The tiles system is extremely *network-centric*, and the demands on the network are very important since the game and communication messages need to be broadcasted from tile to tile. For this reason, it was important to find solutions that could handle high bit-rate communications over the distance between the tiles.

We also tried out different communication possibilities, experimenting with which different technologies would be suitable to use within a swimming pool, both on the water surface as well as on the pool bottom. We learned that Infrared communication (IR) could fit our needs, as it is relatively power-efficient and essentially directional.

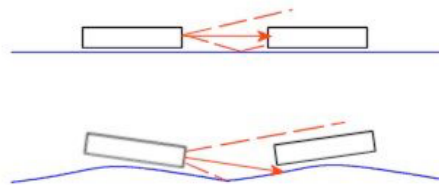


Figure 41. IR communications on the water waves

The IR also proved to work well even while there are small waves on the surface since the IR emitted light spreads quite widely. Another possible constraint could be the extreme sensitivity to ambient light, with transmission range and reliability falling off under certain environmental conditions, such as when there is water reflection and sunlight. This is illustrated in the figure above.

The different envisioned activities seem to be based upon two different communication needs: the direct physical contact between the tiles and ~70 cm distance communication (above water). These seem to cover the needs of the end users involved in the rehabilitation activity. As regard the physical case colored materials should be used instead of transparent ones in order to provide users with

consistent input for the activity. Transparent or blue surfaces may confuse the users' perception as they are too similar in colour to the edges of the pool or are 'hard to detect' due to the reflections in the water.

In the *neighbour-to-neighbour* design the tiles should only talk to other tiles that are right next to them and each side of each tile should be able to carry on a separate conversation with its respective neighbour. The tiles must also be very small. The hardware is implemented in several circuit boards, including the microcontroller board and the input and output boards. The core functionality is on one bigger circuit board and the additional components on a second board positioned next to the first.

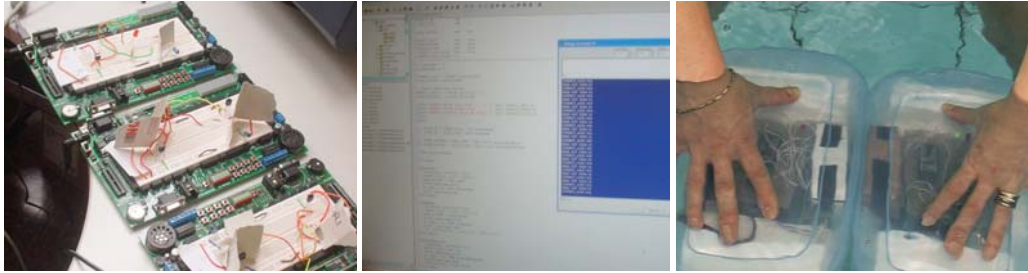


Figure 42. The first release of the lightweight prototype. On the left, the developmental boards, On the middle, the programming environment, On the right, the early testing in the water with the breadboard inside Tupperware boxes.

The first version of the prototype was used in early testing in the water with the aim of exploring communication over the IR channel in the water and a preliminary waterproof solution for the design of the tiles.

These trials allowed the prototype to evolve into a more robust implementation in which visual feedback and autonomous AC battery were added and experimented with. In the second release one high power LED for each side was used for the feedback and a 7.5 V battery was adopted for the power supply. Due to the constraints in the Basic Stamp 2 microprocessor, we limited this prototype to communicate at the most on two of its six sides. A simple communication protocol has been implemented so that the tiles can communicate and understand when they are placed in a correct sequence. When correctly located in relation to the other tiles, a light feedback is provided through a PowerLed that is connected to the micro controller.



Figure 43. On the left, the second release of the prototype. On the Right, the plastic boxes adopted.

With the second release we also tried out forms that could respond to the Active Surfaces concept. Thus, while protecting the hardware in the waterproof Tupperware boxes we put them into square plastic boxes that embodied the form of a tile. We also built these tile boxes in such a way that they were still able to float. In fact, we took care of the positioning of the battery, the most weighting component, and we added polyester to make the objects more robust and compact.

The case needs to be fully watertight but still fairly easy to open, in order to access batteries for recharging as well as to reprogram the micro controller and to allow for changes that emerged during the trials. One important requirement indicated by the therapist is that the patients should be able to take two tiles at the same time by opening their arms, and also to physically snap the devices in order to fix them in a certain pattern. When correctly connected the tiles can give visual feedback through red lights that are easily perceivable in the swimming pool. The prototyped tiles are characterized by the possibility for applying layers on the top where different surfaces and materials can be then be attached as input for the activities. In this way different surfaces can provide different tactile experiences to the users or parts of images can be composed in order to render the complete picture.



Figure 44. On the left, the exploration of materials. On the right, the construction of game surfaces.

The outcomes of the trials with the prototype in the swimming pool indicated that the physical features of the surfaces are strictly dependant on the context of use. At this point we started to investigate the different materials that could be used for the casing, assessing how they worked in water, how they were perceived by patients and therapists in the pool and how they could reinforce the different rehabilitation activities.

We tried out plastics and foam with different consistencies in order to verify the different tactile experiences they might facilitate when put into the water. In fact, not all of the materials give the same tactile feeling inside and outside the water, i.e. when dry and wet.

The exploration resulted in the choice of a special sponge - also used in copy machines to absorb and release ink - and of wooden sticks to create patterns with different textures and shapes. The first two games that have been designed were an image composition sequence based on simple abstract figures (i.e. stars and circles), and a sequence from the thinnest to the thickest textures of wooden sticks. In both the games we tried to work on the visual and tactile experiences of the children in order to provide a broader access for children with special needs, in particular those with sight impairments.

The fourth and last release of the prototype resulted from a user evaluation and a refinement of the previous prototype. We needed to increase the visual feedback to make it more consistent by adding more LEDs and contrasting the light with a black surface.

This early prototyping phase ended with the construction of the initial lightweight embedded prototype that was tested in the actual scenarios of use and evaluated together with therapists and trainers at the swimming pool. The prototype is shown in the figure below.



Figure 45. Initial Lightweight embedded prototype

As reported in this section, the hardware implementation of the initial lightweight prototype followed the inspiration of working with minimal interactive features. In fact, we continued the design process by adding basic units, i.e. single LEDs, to the initial design in order to better fit with the requirements we had, while still maintaining a very simple and lightweight system.

The prototype we developed has been used for early exploratory tests with the targeted end-users, both the therapists and the patients, during the current therapeutic sessions in the swimming pool.

The main activities we designed are based on the creation of sequences through the adoption of the three tiles we have in this version of the prototype. The activities were envisioned in the participatory design session with therapists and trainers, who relied on their current practice and on the expected potentials of the Active Surfaces. The activities presented below describe some of the game explored in the pool with different patients. The tasks have been tailored to the specific needs of the disabled individuals that were children with sight and physical impairments.

Exploration of the Image composition task

The therapist suggested an activity based on image composition. Each tile had a surface attached to its top that was (re) movable in order to setup different games. In this task each of the three surfaces displayed a part of the images that the children had to compose. The composition consisted of placing the tile in the right sequence. The images to be composed were a circle and a star.



Figure 46. Image composition activities utilizing the Active Surface prototype

The parts of the images offered the affordances for the patients to accomplish the task. The side area of the surface showed half of a complete image or symbol. Furthermore, a step-by-step visual feedback was given each time the tile was positioned in the correct order. This was designed in order to provide progressive and coherent feedback during the creation of the sequence.

Exploration of the Texture based task

The therapist used Active Surfaces in order to propose a game based on tactile exploration. This activity involved visually impaired patients. The therapist wanted to stimulate the different sensory channels, providing both specific inputs and appropriate feedback, such as vibration.



Figure 47. Tactile exploration

Thus the therapist attached three different surfaces made of pieces of wood measuring different dimensions on the top of the tiles. Using pieces of different sizes, the textures provided the patients with diverse tactile perceptions. The goal of the activity was to create the sequence from the thinnest to the thickest. When the right tile was appropriately placed in the sequence, it vibrated (and lighted up) to provide the patients with positive feedback.

4.3.1.2 Current Lightweight embedded prototype

The current Lightweight prototype has resulted from the design workshops described in Par. 3.4 and from the software and hardware development of the initial Lightweight prototype described in the previous paragraph.

The prototype consists of four 30x30x6 cm square tiles made of the base of a cap with a hole, as shown in the figure below, and, on top of that, a cap with no hole that constitutes the top cover of the tile. In between the two caps a rubber o-ring is placed in order to make the tile watertight. A frame has been designed on the upper cap with space for the magnets to easily snap the game surfaces. The new game surfaces developed are rigid plastic squares (30x30 cm), which have a magnetic frame that is complementary to the one on the upper case (i.e. the male on the surfaces and the female on the case). In contrast to the initial version of the prototype, with the new game surfaces we mainly concentrated on visual input and image composition games, such as puzzles and domino games. Each surface displays a bright colour contrasted with a black background.

Magnets have also been used to snap the tiles together and to make it easy to take them apart throughout the activity. While floating on the water the tiles tend to move away from each other; it is important that the magnets are strong enough to keep the tiles together and also easy to break apart in order to favour the composition and decomposition of the structures through the activity.

The tiles have strips of PowerLEDs located each side along the edges. The location of the LEDs on the edges was chosen to leave as much free space on the top as possible. In that way the surface can be used flexibly with as many game images as desired. The game surfaces to be applied on top of the tiles have thus been designed to remain visible in spite of the LED strips, e.g. the puzzle games with the light strips that break the shapes created by the surfaces.

In the current prototype the Assembler Tile has also been developed as an early mock-up and working prototype. As it is the means for programming and debugging the tiles, the Assembler Tile has been built as a tile with peculiar features. In fact, it doesn't have to have the same physical features of the common tiles. It is smaller than a common tile (15x20x5 cm) and it isn't necessary that it be

waterproof since it must be used outside the water. It has a processing board and an IR communication board. It constantly reads the incoming messages and has a display in which is visualized the common message information that the tiles send: TileID and SideID. In this way the Assembler immediately shows the incoming messages and can be used to easily debug the ordinary tiles. The Assembler Tile Lightweight prototype has two buttons. One is used to program the tiles sequence by making the tiles enter the programming mode and activate the sides' recognition. In this way the tiles, previously positioned into a sequence by the therapist, come to know who their are neighbours and record this information. A new game sequence has thus been established. The second button is instead used to send a common tile message (TileID, SideID) via IR. This function is particularly valuable for debugging the behaviour of the common tiles and to assess the functioning of the communication channel (Sender/ Receiver).

The Assembler Tile lightweight prototype proves to be useful as a means for allowing the user to experience physical programming-by-example and the dynamic configuration of the system.

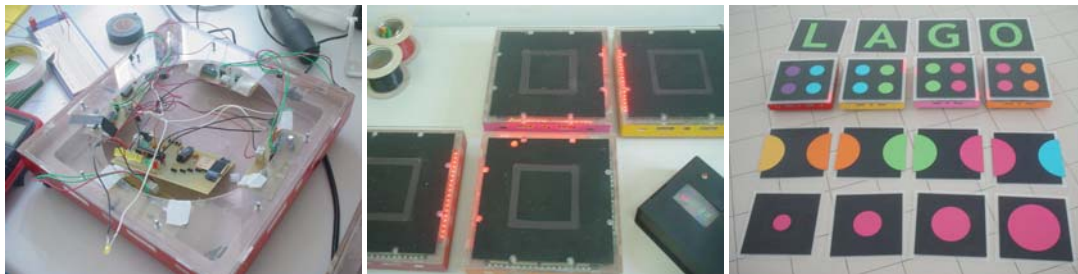


Figure 48. On the left, the hardware into the tile physical case. On the middle, the current Lightweight prototype. On the right, examples of the game surfaces developed together with the prototype.

Together with the development of a new tile case and new game surfaces, the hardware and software have been completely revised as well. In fact, the current Lightweight prototype is a robust system that has been developed by building upon the outcomes of the testing with the initial Lightweight prototype. The current implementation of the tiles uses serial communication implemented over InfraRed with a working frequency of about 38 KHz. The IR-sender is connected through an AND-logic directly to the serial line and a 38KHz signal. This creates the correct pulse-train that can be read by an IR-receiver and used as input directly to the Rx line of a serial port. In the current prototype, up to 6 IR senders/receivers can be used in parallel through a developed serial router developed on an AVR Mega 8 microprocessor.

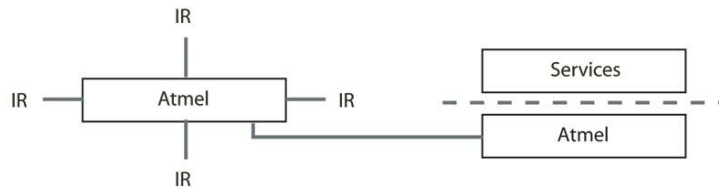


Figure 49. Lightweight prototype stack.

In this case 1 serial port is used to develop 6 software serial ports that have been implemented on the microprocessor. Since software serial ports lack buffers of the ingoing and outgoing traffic, each port is connected to another AVR-chip, the AVR tiny 2313, which enables the buffering of the incoming and outgoing signals. The programming idea is to move from a serial implementation to more Morse

code-like communication scheme where a unique start sequence is used, followed by the data and finally a unique termination string is sent.

The different values are encoded using the time of the IR-bursts to encode the data. E.g: start_bit = 1 timeunit; bin_0 = 2 timeunits; bin_1 = 3 timeunits; stop_bit = 4 timeunits; pause between pulses = 2 timeunits. The reader would wait for the incoming signal, when a signal becomes High (someone is sending) it starts to measure the time units until the signal goes low again (i.e. there is a pause), if the first signal read is not a start_bit, skip all the following signals until a start_bit arrives.

We use two port-registers on the AVR controller to have 8 incoming and 8 outgoing pins to be used for the implementation. Each pin corresponds to one incoming or outgoing signal. We send data as two bytes on the serial port to the decoding AVR chip. The first byte is the outgoing port, the second is the data.

When an incoming byte has successfully been decoded, we send two bytes on the serial line to the main controller, i.e. first byte=incoming port, second byte contains the data.

As for outgoing data, an array should correspond to each pin where the data (i.e. bytes) are stored in binary mode, corresponding to the time units indicated above. Furthermore, the pauses are added to the arrays as 0:s. In each loop of the application, we will parse the first binary value of each outgoing array (FIFO); if there is a value (1 or 0) we add this value to a byte that we construct, if a value is lacking we simply adds a 0 to the byte on the corresponding binary position in the byte.

To each pin of a port we connect an IR-receiver. The IR-receivers force the corresponding pins to go high or low. We will have to read the port-value, all 8 pins at once, and decode the binary values into 8 arrays, one for each pin. We browse the arrays searching for a start_bit, once found we read the following eight binary values. If the next value is a stop_bit, we know that we have read a correct byte and we can send it to the main controller on the serial port.

As already suggested in the above paragraph, Lightweight prototype tools, while required to support evaluation in an actual use context, do not satisfy the main objectives of leveraging the architecture and demonstrating its utility. In fact none of the software applications utilized have been developed from the PalCom architecture.

The importance of both the Lightweight embedded prototypes in this research is that they have been used throughout the process to express some of the key practices for the emergence of palpability. They have been developed ad-hoc to cover some of the relevant issues of the Active Surfaces application that are otherwise hard to demonstrate without the fully developed PalCom architecture. These prototypes can't substitute the PalCom prototype as they are very limited to specific issues of the activity. At the same time, as we will describe in the next chapter regarding User testings, they have been fundamental for giving the feeling of how palpability emerges in the actual context of use. Supporting palpability from the perspective of use, they leverage the potentials of the infrastructure assuming that it makes a critical difference for the users.

The Lightweight prototype allowed us to experiment with issues like:

- Playing and Programming
- Re-configuration
- Debugging
- Exploration

The user testing that we have conducted with the Lightweight prototype are expected to inform about these specific issues. We assume that this process informs the exploration of the Qualities the Open Architecture might have to support palpability.

In fact while the architecture is not adopted in this prototype, it exhibits some use properties that let palpability emerging through the use. In particular we have assumed that:

- the opportunity to directly play and program the tiles informs the *Assemblability* quality,
- the re-configuration would be a pre-requisite for *Adaptability*,
- the debugging by using the Assembler Tile would contribute to *Resource Awareness*, in particular regarding the 1st Order resources,
- the exploration would inform the *Experimentability* of palpable systems.

This process is described and detailed in Par. 4.4.

4.3.2 Architectural prototypes

The development of the Active Surfaces architectural prototypes is the result of the process described in this chapter. The applications developed with the PalCom open architecture have been either deployed within the simulation framework or implemented as physical embedded prototypes.

The Active Surfaces system is one of the PalCom applications that underwent a concurrent development either within the simulation framework and the hardware platform. The PalCom architecture experienced on the simulated framework is likely to inform the development of the embedded applications.

As described above, Active Surfaces represents a *toy problem* for Ubiquitous Computing applications consisting of portable, distributed and small devices to be used in experimental settings. Due to the characteristics of the needed platform, the embedded tiles are affected by the constraints of the current hardware implementation. When the tiles are deployed as simulated devices on a desktop machine they are expected to have an optimal performance and can exhibit the level of experimentability suitable for palpable technologies.

In fact in the Active Surfaces scenario the hardware platform selected for the Embedded Architectural prototype is the UNC20 microcontroller. With such small microprocessor only the PalVM is supported as a runtime engine. The middleware management layer, which consists of managers handling resources, services, assemblies, and contingencies, requires too great a memory footprint to fit into the 8MB memory of the UNC20.

Therefore, the software for the tiles has been developed to run on a standard PC with simulated infrared communication in concurrence with the development of the hardware for the tiles and the optimization of the middleware management layer. On the desktop machine the simulated framework runs on top of Sun's JavaVM.

The adoption of simulated and embedded prototypes also introduce a valid distinction with regard to user experiences. In the Active Surfaces scenario the characteristics of the context and the features of the concept, mainly constructability, configurability, support for creativity, modularity and, moreover, physicality invoke a purely physical exploration of the tiles application. In fact the therapeutic treatment in the water is mainly based on the physical experience and the presence of tangible devices is of remarkable value in these practices. In spite of this, the use of the Simulated Architectural prototype contributed to both the architecture development and the exploration with the users. In fact the therapists had the valuable opportunity to use the architecture, especially the middleware managers and the applications, even if within the simulation framework. In this way they have experimented of the future use of the embedded tiles would be. The details of the experiments with the Simulated prototype are described in the next chapter.

In the following two paragraphs the Architectural prototypes, the Simulated prototype and the Embedded prototype, and the open architecture that they utilized will be described in detail.

4.3.2.1 The Simulated Architectural prototype

To support the development of game logic and software for the tiles a simulation framework has been developed. Having a software simulator available makes it possible to develop the architecture and the hardware in parallel. Specific architectural components that have been not available for the embedded platform can be used in the simulated framework for debugging and profiling (Bronsted et al. 2007b).

The Active Surfaces Simulator Architectural prototype consists of a model of the swimming pool as an environment for infrared communication, and a graphical user interface (GUI) for the manipulation of the physical location of the tiles. Each tile is a simulated PalCom Device that can be assembled with other Devices and Services. The GUI consists essentially of four classes: the Pool class, which simulates the pool in which the tile floats, the PoolPanel, which displays the floating tiles, the TileImage, which generates the various game surfaces of a Tile and the TilePeer, which is responsible for rendering a Tile. As with the embedded prototypes, the physical assemblies (i.e. the sequences) and the game surfaces can be both changed and configured with the tiles simulator as well. In fact, the assemblies are created by using the simulated Assembler Tile and the game surfaces can be easily adapted by modifying the TileImage service and thus loading different game images on the simulator.

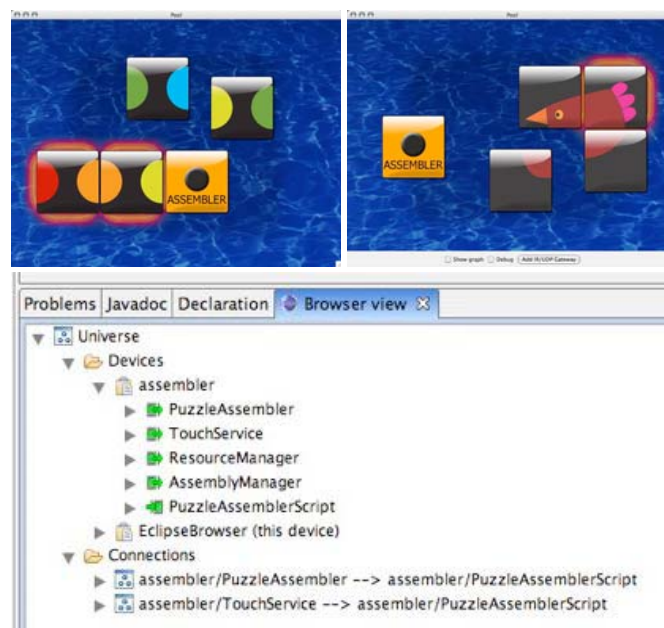


Figure 50. Up, left and right, the user interface displaying different games. Down, the browser view of the Assembler Tile on the Eclipse environment.

The user interface is connected with the pool model so that when a tile is moved in the user interface, the pool model is updated accordingly. The model of the pool is also used in the Media Abstraction Layer (MAL) of each of the tiles; this component is described in Par. 3.3.4. When a tile sends a message, the Media Abstraction Layer of the tile accesses the pool model to determine which tiles it is connected to and delivers the message accordingly.

When the simulator is launched there is no connection with the network. To connect one of the tiles to

the Eclipse Browser it is necessary to attach the IR/ UDP gateway and select the tile that should be connected. In fact, the simulated tiles communicate via IR while the browser uses UDP protocols. To be able to talk to each other a gateway between the IR port of one of the tiles and the UDP interface needs to be present. The services running on the Active Surfaces simulated framework are the ConnectivityService, which is responsible (with the ConnectivityUpdateEvent) for keeping track of the connections and sending commands when a connection is lost or a new one appears, the LEDService, which controls the LEDs on the tiles and the TouchService, which together with the OneTouchEvent is specifically developed for the AssemblerTile and indicates that the button on the AT has been pressed.

The connectivity is also managed by the CommunicationThread component, which delivers messages based on information about link connectivity from the Pool and keeps track of connectivity information to be used by the services. The ConnectivityModel is instead deputed to manage the game logics by getting SideHappy and GlobalHappy messages. While being moved within the pool in the GUI, a tile exhibits the light feedback when it establishes a connection with the other tiles: the simulated LEDs light up on the connected side. When the tile is globally happy all the sides light up and when the winning solution is achieved, the whole sequence of tiles starts blinking.

Because it is a distributed set of networked devices, the processes occurring among the components also need to be scheduled by the PalcomScheduler.

The figure below shows which components of the architecture are realized and utilized for the Active Surfaces simulated prototype.

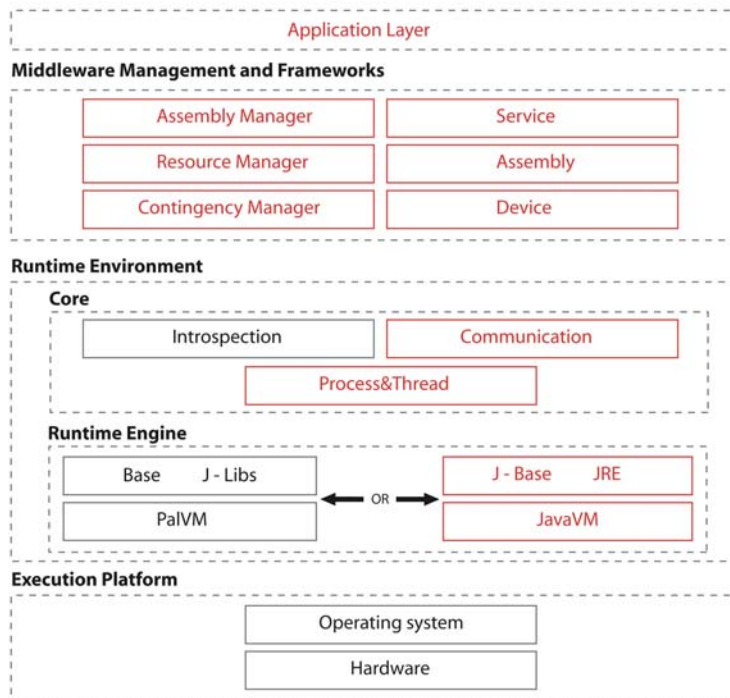


Figure 51. The open architecture realized in the Active Surfaces simulated prototype.

The Simulated architectural prototype allows for the experimentation with prototypes of core architectural components, for example the PalCom Assembly.

The Assembly architectural prototype has been developed to define how a set of distributed and networked components are connected and how they work together. In the Active Surfaces application the assembly prototype has been relevant to understanding how the users could orchestrate

components, services, and connectors in a dynamic, contingent way. By prototyping the Active Surfaces assembly within the simulation framework we developed an architectural *toy* that users can immediately understand and play with. In fact, the simulated prototype has a videogame-like appearance through which the users can directly experiment with a purposeful aggregate of tiles. Furthermore, the testing that involves the repeated rearrangement of tiles is much easier done by using a graphical user interface than by physically moving the actual tiles around in the water.

The Simulated Architectural prototype allowed us to experiment with issues like:

- Playing/ Programming
- Creation of assembly
- Game management
- Exploration

The user testing that we have conducted with the Simulated prototype are expected to inform about these specific issues. As being an architectural prototype, it utilizes part of the Architecture and realize the Qualities for palpability. We assume that the issues outlined above inform the exploration of the Architectural Qualities in the following ways:

- the opportunity to directly play and manipulate assembly of tiles realizes the Assemblability quality,
- the creation and configuration of assemblies realizes Adaptability as User Adaptation,
- the game management by using the Assembler Tile would contribute to Resource Awareness, in particular regarding the 2nd Order resources,
- the exploration would inform the Experimentability of palpable systems.

4.3.2.1 The Embedded Architectural prototype

The second Architectural prototype consists of a set of four embedded tiles, consisting of both commercial hardware and hardware parts designed ad-hoc and then built at the University of Aarhus within the project. Like the Lightweight prototype, the Architectural tiles can also be connected to each other to form a network and support multiple games by having a multi- purpose programmable hardware. The Architectural prototype is characterized by small form factor and low power consumption, which is particularly suitable for palpable devices.

At the beginning of the process, before the hardware implementation, a technical simulator on top of the Corundum experimental framework was developed. It allowed for the development and testing of a flood-routing algorithm for the ad-hoc network made up of the tiles, and for the implementation of simple versions of the *Follow-the-Path* game, described in Par. 3.1.3. It has also been useful for testing the assemblies. A set of basic services encapsulates the basic hardware functionality of the tiles and each game is implemented as one or more unbound services that can be connected to these services.

The main challenge for the architecture development is related to the networking mechanisms that have to run on small embedded devices with an extremely low bandwidth and limited processing power. This is a resource-constrained device that can allow about 2Mb for each application. As discussed in Par. 3.2, both horizontal and vertical scalability are core issues for the Active Surfaces application. The number of messages exchanged among the devices, as well as the performance of the routing mechanisms increases with the number of tiles in the game. This also depends on the actual game logics and how they are implemented.

The tiles communicate through their sides via infrared in all 4 directions, and thereby establish an ad hoc network when they are in close proximity. These connections are low bandwidth and short range, but are able to support the tiles while detecting neighbouring tiles and determining their configuration. Due mainly to performance problems and the unreliable low-bandwidth infrared communication, the prototype does not utilize the entire PalCom communication stack. In order to have a better performance, the current implementation bypassed the PalCom communications module.

As already introduced in Par. 3.4.2.1, the main computational unit of the Embedded tiles is the UNC20 module, which is an ARM7 based embedded system running uClinux (uClinux: <http://www.uclinux.org/>) at 55MHz with approximately 8MB ram.

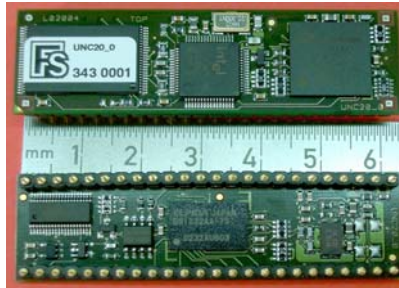


Figure 52. The UNC20 main board.

The primary processor stably runs the PalVM on top of uClinux (see Par. 3.3.4.1), implements the higher-level protocols: packet transfer, routing, etc., and supports the dynamic download of new services, e.g. the Game Services.

Inside each tile an embedded system uses infrared light to communicate with and detect the presence of other tiles. Two tiles can communicate only if they are close to each other and the bandwidth of the infrared communication is approximately 600 bits per second. On each side of the tiles light emitting diodes (LEDs) provide visual feedback to the user.

The UNC20 module communicates with a sideboard through a serial connection. The sideboard is responsible for controlling the infrared communication and the LEDs. The communication board is a microcontroller based IR communication and sensor board. It has a DLP-245PB module with an USB interface via FTDI chip and Microchip PIC16F877 modulator controller (1K flash ROM, 72 bytes RAM, internal 4 MHz oscillator) for IR modulation. The PIC16F505 receives PWM signals on 5 ports, and modulates them onto a 38 kHz carrier, and drives the IR LEDs with this output. Such modulation requires extremely tight timing, i.e. 38 kHz allows 26 instructions at 4 MHz. The reception and demodulation is handled by an integrated HIM602 IR receiver that provides the demodulated PWM signals to the PIC16F877 controller.

Each IR module consists of a couple of close IR emitters and receivers. The sending IR from a LED adjacent to the receiver will saturate the receiver, so the IR communications is half-duplex.

The UNC20 and the Communication modules communicate via a RS232 serial 3-wire connection at 9600 bps. From the outside the communication module seems to have 6 ports: 4 IR ports (I/O), 1 LED port (output) and 1 touch port (input). The UNC20 board changes the serial line levels from 0-5V to -12 – 12V to provide a standard RS232 interface.



Figure 53. The UNC20 and the communication module.

The tiles have to be outside of the water in order to both charge the tiles and program/configure a game. An Ethernet connection is required between the microprocessor and a desktop machine running the PalCom software to download services and launch them on the embedded device via a Terminal application.

The most prominent characteristic of the Active Surfaces embedded architectural prototype is the very transient ad-hoc networking. This has put great demand onto the PalCom routing and discovery mechanisms. Further optimization for speed and possibly space is needed if this embedded Linux device is to handle the entire PalCom communication / discovery stack. The architects were aware of this trade off, and have periodically improved the PalVM performance throughout the process. In this architectural prototype the management of the IR communication is split into two parts in the Base and in the Communication component. The actual IR code that resides in the Base communicate with the hardware, while within the Communication component the IRMediaManager handles the sending and receiving of the PalCom messages over the PalCom protocols, relying on the code in Base. The figure below shows which components of the architecture are realized and utilized for the Active Surfaces embedded prototype.

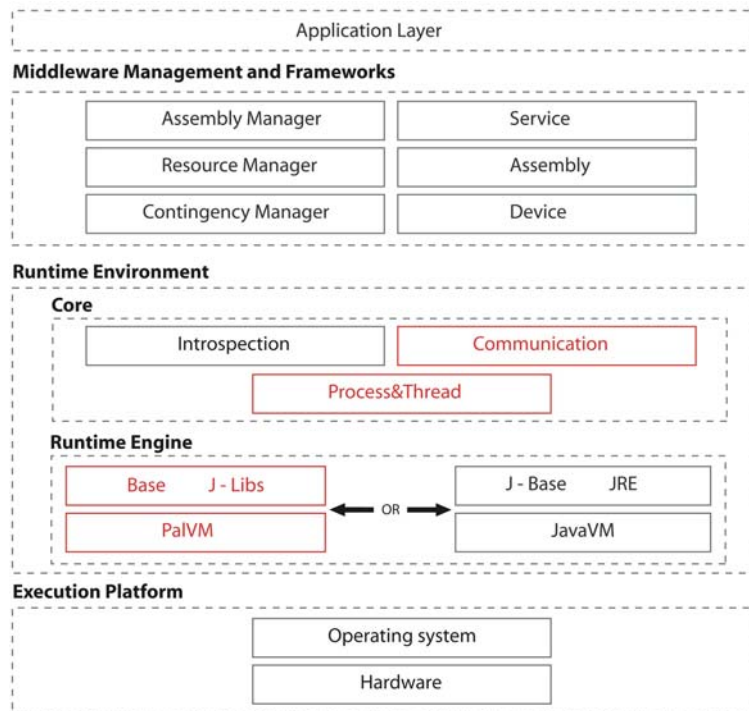


Figure 54. The Open Architecture utilized in the Embedded architectural prototype.

The Embedded Architectural prototype allowed us to experiment with issues like:

- Communication and Discovery
- Game Setup and Re-configuration
- Performance

The performance testing that we have conducted with the Embedded prototype are expected to inform about these specific issues and, directly, some Qualities for palpability. As being an architectural prototype, it utilizes part of the Architecture and realize the Architectural Qualities in the following ways:

- the communication and discovery among the tiles inform Resource Awareness,
- support for game setup and configuration is a precondition for Adaptability as User Adaptation,
- good performance is the enabling condition for Experimentability.

4.4 Prototype Scenarios

According to the design approach followed in this research, which was described in Par. 1.3, the scenarios are the tools that trigger the design process as well as a means for evaluating design solutions. Indeed, scenario generation can be considered as an ongoing practice that continues through each iterative cycle of the process.

At this stage of the project, in which the prototypes are developed, scenarios play a fundamental role. They have the aim of integrating the prototypes with the current therapeutic practice; moreover, Prototype Scenarios will specifically address the next design phases, triggering concept refinement and the fine-tuning of the envisioned and prototyped solutions. Prototype Scenarios represent detailed interaction paths, and allow the refinement of the interaction modalities and the physical and material features of the prototypes (PalCom Deliverable 17).

Throughout the scenarios development participatory sessions have been carried out where potential users took part. People who are not involved with research but who are potential users of the technology often have a very different view on the technology developed.

By putting the concept into a prototypical representation, either with a working prototype or video representation, the imagination of the counterparts was stimulated and led to fruitful discussion and further development.

First, the users received an introduction to the overall concept, and then the prototype was demonstrated. People were allowed to discuss with researcher the prototype and its functionality. The discussion centered on every day scenarios of the therapist, imaging how to the technology may be introduced in their context. As potential users imagined the technology to be used in their context, questions about eventual functionalities came up and where given to the researchers (e.g. “Could the tile know that they are in the water? Could the Assembler Tile take pictures as snapshot of the activity?”).

Based on these questions the users helped on constructing the following scenarios that thus have value to them. These discussions were very fruitful for the researchers and for the users.

The outcomes have been summarized in the Prototype Scenarios with the aim of better understanding the technical requirements, the physical appearance as well as the role of Active Surfaces in supporting pre-existing or emergent activities. The binding with the user activity is also provided by the strong connection the Prototype Scenarios have with the Activity Scenarios, described in Par. 2.4.

They resulted from a further elaboration of the Activity Scenarios through which we try to see how Active Surfaces may impact on the key therapeutic Practices.

Eventually, the scenario generation process would also initiate a new design cycle in which new concepts as well as new design perspectives and settings might be taken into account.

Prototype Scenarios are used for detailing the future therapeutic practices with the use of prototypes that have not yet been developed in terms of function, role or look-and-feel. This use of Prototype Scenarios aims at completing the design process that has been carried out. The scenarios are informed by the results of the analysis that is carried out for the development of the current prototypes and they contribute to the development of the future prototypes. In fact, by having the scenarios evaluated with users it is possible to merge the previous and the future cycle of the design process.

The existing Active Surfaces architectural components are described together with those aspects that would be implemented in the future release. The Prototype Scenarios are thus the means for the evaluation of the current prototype and also the trigger for the possible future development of the architecture and the application.

The Active Surfaces concept and prototype have been experimented with in the Prototype Scenarios. The Lightweight and the Architectural prototypes tend here to converge toward a unique vision of the Active Surfaces system. It embeds the architectural Qualities of Assemblability, Adaptability, Resource Awareness and Experimentability through the adoption of the PalCom open architecture.

As described above, each of the prototypes we have developed and experimented with embody the Architectural Qualities in several different ways. At this stage we consider a unique Active Surfaces prototype that summarizes all the features of the application. The figure below shows the parts of the Architecture utilized by Active Surfaces and how the Qualities can be primarily ascribed to them. Even if the Qualities generally depend on the behaviour of whole Open Architecture and don't emerge alone, they can also be put into direct relation with single components.

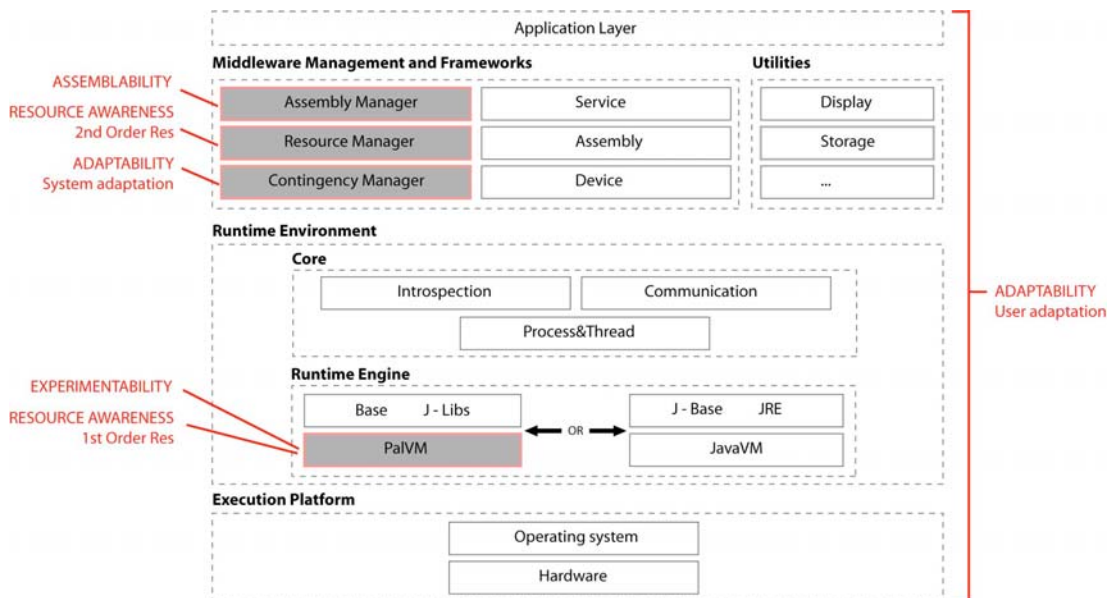


Figure 55. The Open Architecture and the Qualities

In what follows we try to summarize the characteristics of the Active Surfaces prototype as they emerged throughout the prototyping.

We have translated the Qualities into a set of more concrete issues that may more easily support the process. Active Surfaces is thus described in relation to the following Qualities-related issues:

- the opportunity to directly play, manipulate and program assemblies of tiles, which informs Assemblability,
- support for game setup and re-configuration of assemblies, and how this achieves Adaptability as User Adaptation,
- the communication and discovery among the tiles, the debugging of 1st Order resources, the game management of the 2nd Order resources and how it informs Resource Awareness,
- good performance and exploration and how it informs the Experimentability of the system.

Prototype Scenario 5

Creative construction (Assemblability)

Alessio is planning the therapeutic session for tomorrow. He is in his room at the rehabilitation centre and he is focusing on the objectives of his patient Andrea.

He is now configuring a new game with Active Surfaces to support physical activation and cognitive stimulation. The therapist wants to setup a chess-like game and he produces the chess piece images (e.g. the King, the Queen, the Tower, etc.) as Tiles' surfaces, i.e. in a rigid plastic format to be folded in the upper tiles. Before the beginning of the session Alessio decides to start with four tiles; as soon he constructs the sequence, the tiles dynamically recognize each other.

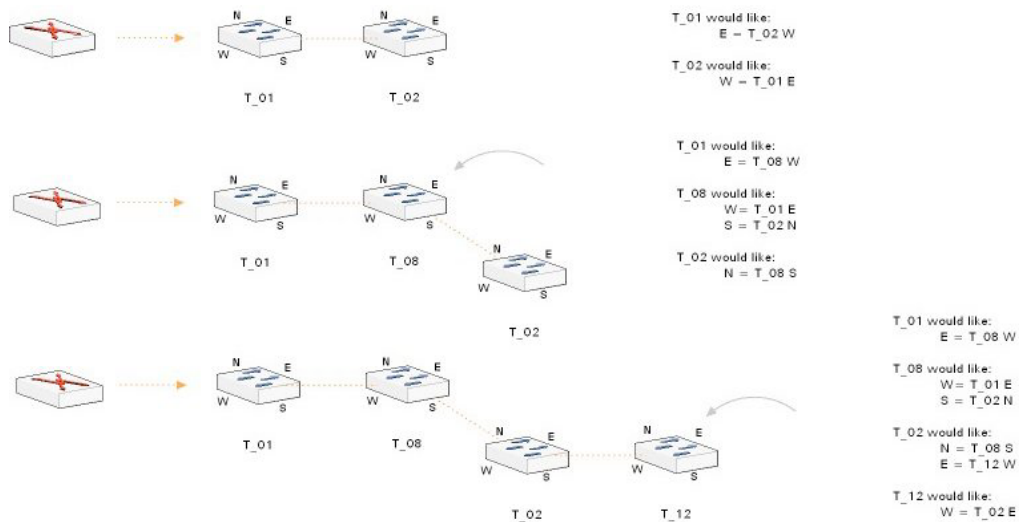


Figure 56. The tiles dynamically recognize each other

By arranging the physical assembly of embedded devices Alessio is also preparing to initiate the (software) tiles assembly.

He uses the Assembler tile to define which Services to adopt - for example the VibrationService as output - and to establish the assembly. He creates both the assemblies of services in each Tile and the general assembly made of each Tile assembly while creating the game.

He creates Assemblies in a physical programming-by-example, by physically showing the tiles the winning sequence.

The four Active Surfaces tiles are ready to be used. Alessio has defined the rules of the chess like game and which tile wins over the other. He will try with a competitive game like chess that is particularly appreciated by the child.

Prototype Scenario 6

Dynamic adaptation and re-configuration (Adaptability)

Laura is in the water with Federico, a 7-year-old boy with Autistic Spectrum Disorders. From the beginning of the session Laura tries to maintain a dialogue with the child. She starts to use Active Surfaces and pushes the tiles against the other.

Federico starts to laugh; he seems to appreciate the activity and slowly starts to imitate the therapist. After few minutes, as soon as Laura and Federico move the tiles, they splash each other, and the child starts to cry and shout with discontent: he really dislikes being spattered with the water.

Laura abandons this game and waits for a minute. Then she starts the DominoGame assembly by launching it with the Assembler Tile. Laura carefully introduces the tiles and the Domino game surfaces. Federico enjoys the colours of the surfaces. After 3-4 minutes the child slowly loses interest in the game and moves away.

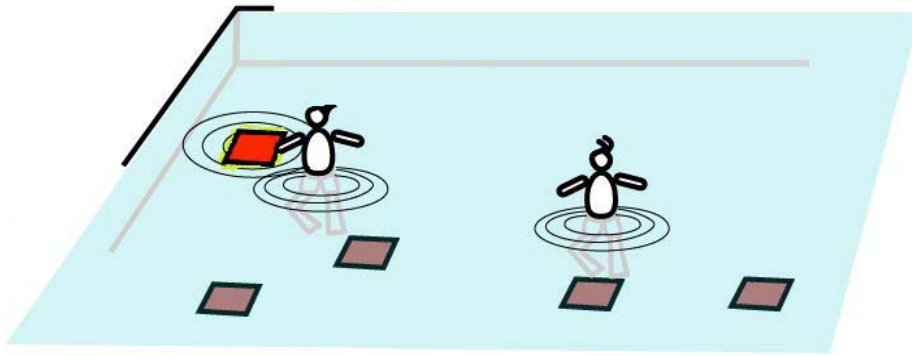


Figure 57. The child has to pick up the blinking tile in the LightUp game

After 5 minutes Laura decides for a new game. By using the Assembler Tile, she configures the set of tiles she wants to use for the LightUp game. It consists of the tiles alternatively blinking one after the other. The light captures the attention of the child. He follows the light and reaches each blinking tile by moving within the environment. He is clearly engaged in this game and Laura lets him playing freely.

Prototype Scenario 7

What's going on (Resource Awareness)

Silvia is using Active Surfaces together with Lucia, an 8-year-old girl with cerebral palsy. She has started the Puzzle game assembly: each tile is a piece of the puzzle. While aiding the child to complete the puzzle the therapist notices that one of the tiles is not properly working. Even when correctly placed among the other tiles this tile doesn't give the proper light feedback. Silvia uses the Assembler Tile to assess the behaviour of the tile. She aligns the two tiles and makes the Assembler read the IR messages coming from the tile.

[1]

She sees that the receiver is working properly as the tile lights up on the correct side, but determines that the problem is in the emitter, since it doesn't send any message. Shortly thereafter Active Surfaces are again working to a full extent. Silvia has substituted the degraded tile by introducing another tile and creating a new Puzzle game assembly.

[2]

She sees that both the IR emitter and receiver are working properly, the tile recognizes the neighbouring tiles and also announces itself. Silvia tries the other side connections out and sees that some are not working properly. It doesn't seem to be a problem of the hardware components but rather a malfunction of the ConnectivityService, which even when degrading still guarantees part of its functionality. Silvia "calls" the ResourceManager via the Assembler Tile and it confirms her hypothesis. Silvia understands what is happening and the ResourceManager immediately tries to recover by automatically re-starting the corrupted service. One minute later Active Surfaces is again working properly.

Prototype Scenario 8

Handling technology (Experimentability)

Elena is now using the set of Active Surfaces tiles for the intervention with Enrico, a 10-year-old child with a mild mental. As soon as the child sees the tiles he starts to throw them around. The tiles don't lose their assembly configuration and preserve game instructions time.

Elena let him different formations of tiles, connecting them as if they were bridges, towers and boats. Many possible combinations allow the symbolic use of the tiles together with other toys.

The light feedback that the tiles contingently placed in the correct way captures the attention of the child. Enrico holds one tile and starts to use it as "tile detector" to find the right tile connections. He follows a try-and-error strategy and, step-by-step, he is able to construct the tile assembly.

The Active Surfaces system described in these Prototype Scenarios has been implemented in the current different experimental prototypes. Both the outcomes of the experiments with the current prototypes and the use of Prototype scenarios are expected to provide indications for the re-design and future development of Active Surfaces.

The figure below represents the relation between the Activity and Prototype Scenarios and the Architectural Qualities. These can be thought of as constituting the backbone we use to describe the design process as well as the experimentations. In fact, the Prototype Scenarios presented in this section are also used to describe the experimental trials of Active Surfaces.

In some cases, while the Architectural prototypes have not been currently developed, we have used the Prototype Scenarios to envision how the open architecture would fit with the requirements from these scenarios. Currently existing architectural components could be adopted in the Active Surfaces to enable the emergence of novel practices of palpability. The Prototype Scenarios discussed and evaluated with the users will inform the development of these components that is to follow.

ACTIVITY SCENARIOS	USE PRACTICES	PROTOTYPE SCENARIOS	ARCHITECTURAL QUALITIES	PROTOTYPES
1_ Creatively structuring novel games	Looking for creative solutions	5_ Creative construction	ASSEMBLABILITY	Simulated Architectural
				Lightweight
2_ Continuous negotiation and adaptation	Dynamic configuration of the tools.	6_ Dynamic adaptation and re-configuration	ADAPTABILITY	Embedded Architectural
				Simulated Architectural
				USER ADAPTATION
				SYSTEM ADAPTATION
3_ What's going on	Resource availability and opportunities for action.	7_ What's going on	RESOURCE AWARENESS	Lightweight
				1st ORDER RES
				2nd ORDER RES
				Simulated Architectural
4_ Handling materials	Use and performance.	8_ Handling technology	EXPERIMENTABILITY	Embedded Architectural
				Simulated Architectural
				Lightweight

Figure 58. Mapping among Activity Scenarios, Therapists' Use Practices, Prototype Scenarios, Architectural Qualities and Prototypes.

(* System Adaptation is not currently implemented. It will be further discussed in Chapter 6)

Chapter 5

Experimentation

5.1 Introduction

As described throughout the previous chapters, this research follows an interdisciplinary approach that brings together software development, laboratory and naturalistic experiments and participatory design to inspire, inform and evaluate architectural design.

This poses methodological challenges for the design of Ubiquitous Computing technologies. Some well known methods have been adopted, such as Wizard of Oz and other creative techniques, other have been newly created within the PalCom project, like the Traveling Architects meeting.

Within the Active Surfaces scenario we have adopted observational studies and quantitative analysis in order to contribute to architectural design.

As described in Chapter 2, ethnographic observations were used at the beginning of the research to investigate the field. Then, throughout the experimentation phase, both user testing and performance testing were used to discuss Ubiquitous Computing architecture. While empirical studies and participatory sessions provide large returns for the quality of the code for middleware and software architecture design, quantitative measurements on specific components can also directly inform this activity in a more stringent manner.

The Active Surfaces application prototype addresses specific methodological challenges that can be generalized to other projects in Ubiquitous Computing (PalCom Deliverable 33). The therapeutic setting is a domain in which there do not exist practices of the use of ubiquitous computing technologies running on existing software architectures to which they could be compared. Thus it is not easy to predict successful new practices that may emerge in interaction with new technologies. Since integration among technologies, practices, environments and human actors is a complex and dynamic process, it is not easy to anticipate what the impact of the design of new technologies might be. To do so, we conducted empirical explorations of the architectural and non-architectural prototypes described in Chapter 4.

The main purpose of the Active Surfaces prototypes is to enable the iterative development of the PalCom open architecture, and not to fully support new ways of working in a particular domain. The main challenge we had to face, especially with the User Testing, was that the use of experimental and *wrapped* prototypes had to be realistic enough to challenge the underlying architecture and to allow the users to develop new practices of making computing palpable. The prototypes don't always have the robustness and the evocative power to support these two processes to a great extent. To address these difficulties we have developed a range of experimental activities that combine observation, testing, analysis and evaluation, allowing the Active Surfaces to contribute to the architectural development. Together with the data sessions, the design workshop, the Wizard of Oz and the Traveling Architects meeting we have already described, we have carried out many demonstrations and experiments with the different Active Surfaces prototypes. Preparations for, and demonstrations at major events provided opportunities for field observations, training, testing, evaluation and refinement, while experiments required the definition of methods and procedures that have to be meaningful and informative to the architectural development.

The evaluation of Ubiquitous Computing architectures is an emergent domain at the intersection between Software Architecture Engineering and the paradigm composed by Ubiquitous, Context-Aware, Pervasive and Ambient Computing. Each specific field seems to require a dedicated set of methodological and evaluation tools. For example, in context-aware computing, the context acquisition systems are often evaluated using methods known from AI and neural networks; whereas the user interfaces are evaluated with HCI methods (Schmidt 2002).

Recently there has been a growing interest in understanding specific evaluation problems that arise from the use of Ubiquitous Computing systems (Scholtz, Consolvo 2004; Dey et al. 2002). Sholtz and Consolvo point out the peculiarity of computing that is distributed throughout the physical world and how it might impact individual and social behaviours, such as user acceptance and behavioural change, impact on the environment. For evaluation criteria they assume the user attention (focus and overhead), the adoption of the system (value and availability) and the qualities of the interaction (physically embeddedness, dynamic input/ output, multiple devices, multiple users). Then they also discuss criteria related to the use and the person, such as understanding, control, accuracy, appropriateness, and customization.

The experiments with Active Surfaces aim at evaluating the enhanced user experience and the efficiency or stability of the current implementation of the application within the framework of Palpable Computing. This is done mainly by adopting the Architectural Qualities as *backbone* for the experimental design and evaluation framework.

PalCom software architecture provides a set of functional and use-oriented requirements for evaluating composition of devices. Throughout user experiments may try to compose and recompose devices and services by using the different embedded and simulated Active Surfaces prototypes described in the previous Chapter.

The assemblies are also defined from a user’s perspective, and the role they can play and for which applications will be clarified through empirical investigation. As regards developers, they can take control of interaction by deploying components in a scripting language that in principle is contemporarily a simple and powerful solution. As for end-users, assembly composition may be carried out without any programming skills whatsoever. Assemblies are designed to be an intelligible set of tiles that only require intuitive interaction skills; basically physical construction and sequence composition. They also have to be opaque in the sense that they have to remain efficient tools for people who are pursuing their ultimate goals. In interacting with the tiles, both therapists and patients may enact physical interaction modalities, the former through physical programming by example, the latter by manipulating, moving and combining them and this involves elements of interactivity, such as recognizing whether or not they are correctly connected, the IDs and the feedback.

The two focuses, the use and the architecture, require different methods and evaluation procedures. All of the Active Surfaces prototypes chosen span a range of features and have been used in the research for the on-going exploration of the architectural Qualities. This is detailed in the table below:

		<i>Performance Testing</i>	<i>User Testing</i>	
		AS Architectural Embedded prototype	AS Architectural Simulator	AS Lightweight Embedded prototype
Assemblability			X	X
Adaptability	User adaptation	X	X	X
Resource awareness	1st Order Res	X		X
	2nd Order Res		X	
Experimentability		X	X	X

Table 59. The Architectural Qualities mapped in respect to the prototypes and the experiments.

The architecture evaluation allows us to compare and identify the strengths and weaknesses in the architecture in relation to the Active Surfaces application prototype, which would also help developers assure that the architecture is able to fulfil the Qualities in practice.

As detailed in Par. 4.4, there is one Quality, Adaptability as System autonomous adaptation, which has not yet been implemented at any level by the Active Surfaces prototypes.

The architecture component that is directly responsible for this Quality, i.e. the Contingency Manager, is not currently adopted in any existing Active Surfaces prototype implementation. It is available and tested in a number of PalCom application prototypes, e.g. RASCAL (Ghizzioli et al. 2007) and Major Incidents Overview (Kyng, Kristensen 2007)

However, though it has not yet been implemented, System adaptation is very relevant to the Active Surfaces application because of the valuable user and system opportunities it might provide. The potentials this Quality offers to the Active Surfaces scenario have been described and explored by means of Prototype scenario (7), see Par. 4.4. This Scenario, which has been evaluated with the users, contributes to the future development of the architecture and application and it will be further discussed at the end of the Chapter.

In this research we combined the use of scenarios and the use of simulated and embedded prototypes to address the challenge of experimenting with novel theoretical concepts, system designs and deployed systems. We adopted a combination of two different approaches to architecture evaluation:

- Simulation-and-prototype-based approach (Bosch 2000). Simulation-and-prototype-based evaluations rely on a high level implementation of some or all of the components in the software architecture. The simulation is used to evaluate requirements such as performance and correctness of the architecture. Simulation can also be combined with prototyping, thus prototypes of the architecture are executed in the intended context of use. The prototypes described in Chapter 4 have been a central tool while conducting the research that preceded this thesis. Many prototypes have been built and a number of them have been evaluated in greater depth.
- Scenario-based approach (Kazman et al. 1994). The scenario-based architecture evaluation aims to evaluate a particular architectural quality by creating a scenario that very concretely describes the quality-requirement: the Quality scenarios. The Quality scenarios are used to step through the software architecture and the consequences of the scenario are documented. The use of such scenarios is described in Par.1.3.

Prototypes have also been used as tools to communicate with potential users. Even if they are incomplete or sub-systems it has been useful to deploy parts of the system that are already functioning. The potential users, therapists and trainers had the opportunity to try out the technology in their practice and assessed its impact on everyday tasks. This was possible even if not all functionalities had been implemented yet.

The usage of prototypes in architectural development provided the opportunity to have intermediate embodiments of the systems' functionality. Prototypes of software components with different levels of accuracy and completeness have been used throughout the process and they were not supposed to represent any final or complete stage, not even the current implementations represented by the Simulated and Embedded architectural prototypes.

The Architectural Qualities also constitutes the means for synthesizing the different results acquired from prototypes experiments. In fact, we have carried out User Testing with the Lightweight embedded prototype and with the Simulated architectural prototype, for which we conducted empirical observations. We also did Performance testing on the Embedded architectural prototype, measuring specific indexes. In this way we gained both Qualitative and Quantitative data that informed the evaluation of the PalCom open architecture.

The approach taken in the present research is thus both analytical and empirical. The empirical data regarding the qualitative and quantitative assessment of the architectural and non-architectural prototypes are also reported on in the following paragraphs. The objectives (Par. 5.2) and the different experimental trials (Par. 5.3 and 5.4) are described in the following paragraphs. They contain the methods, the experiments and the results. At the end of the chapter the results will be discussed (Par.5.5).

5.2 Objectives

The experiments with Active Surfaces have the overall aim of evaluating the PalCom architecture within the domain of the therapeutic play in the water. These experiments allowed both a laboratory and a field exploration of issues that are directly related to the Architectural Qualities/ Use Practices addressed by this research. The goal is to explore the Architectural Qualities for palpability *in practice*.

This objective is then articulated in two operative objectives:

(1) Evaluate the performance of the PalCom open architecture in resource constrained embedded devices. Performance is especially important in systems that target small devices. It refers to the speed, the accuracy and the robustness of the system in solving tasks based on the Active Surfaces scenario. The objective is to understand whether the architecture has restrictions in and of itself or if and how it is limited by the current implementation. With relation to this we want to compare different releases of the architecture developed through the PalCom project (PalVM-release and PalVM-debug versions)

(2) Explore the use of Active Surfaces in real settings with end users. The field exploration of Lightweight and Architectural prototypes with users in real settings intends to provide a conceptual exploration and refinement of the Qualities and thus a further specification of the open architecture.

5.3 Objective 1 _ Performance testing

5.3.1 Methods

The conceptual framework for Palpable Computing (PalCom Deliverable 37) states that *palpability* is not readily a measurable feature of a system. In fact it is not a property of a system, but something that arises *in interaction with* systems, people and materials. Even though a system cannot be properly labeled *palpable*, some qualities of the interaction make it palpable for the users.

Thus the Architectural Qualities have been translated into objective measures in order to evaluate the performance of the PalCom open architecture in resource constrained, embedded devices (Objective 1). The open architecture aims at making computing palpable and, as already suggested by Bardram (2004; 2005), it is worthwhile to establish measurements on the open architecture that are related to the Qualities for palpability.

Statements like “the system has high performance” or “the system fits well” are acceptable in the early stages of the architectural development process. As more information about the application prototype becomes available, and the first design solutions are outlined, the above statements become meaningless because they are unspecific and non-informative (cfr. Bass et al. 2003). Performance in this research concerns with how long it takes the tiles system to respond when an event or action occurs.

The goal of this experimental phase is to describe the behaviour of the Active Surfaces system in a tangible way by measuring the behaviour of the system. The Qualities scenarios help in describing the performance

in terms of more informative statements or in short technical scenarios. An example from Bass (2003) is “when 100 users initiate ‘complete payment’ transition, the payment component, under normal circumstances, will process the requests with an average latency of three seconds.” This kind of statement allows *quantifiable arguments* about a system to be made.

Writing such detailed statements is only possible when relevant requirements and architectural components have been identified and prototypes have been developed. In particular the Embedded architectural prototypes (see Par. 3.4.2) are used to observe, explore and evaluate the Architectural Qualities that are measured at run-time, i.e. throughout the performance testing.

Qualities Scenarios that describe Performance testing begin with a request for some service that arrived in the system. Satisfying the request necessitates resources to be consumed. While this is happening the system may simultaneously service other requests as well. The response of the system to a stimulus can be characterized by:

- latency (the time between the arrival of the stimulus and the system's response to it),
- deadlines in processing (a mandatory processing deadline),
- the throughput of the system (e.g., the number of transactions the system can process in a second),
- the jitter of the response (the variation in latency),
- the number of unprocessed events (the system was too busy to respond, and the data that was lost because the system was too busy) (Bass et al. 2003).

The measures we have identified refer to the Qualities for palpability realized by the Embedded architectural prototype, see Par. 4.4.2:

- Resource Awareness, referred to 1st Order resources
- Adaptability as User adaptation
- Experimentability

The experiments focus on measurable arguments related to these Qualities. The experimentation with the embedded PalVM aims to gauge the adequacy of the proposed architecture to the Qualities, by directly measuring or analyzing specific tasks.

In particular the experimental tasks can be grouped into the following areas that, we assume, inform the Qualities:

- *Communication and discovery* informs Resource Awareness
- *Re-configuration* informs Adaptability
- *Performance* informs Experimentability

The objective is to analyse the performance of the PalCom architecture, essentially the PalVM, on resource-constrained microprocessors. Together with specific technical goals, the architectural measures allow for a more objective evaluation of the whole palpable system. Objectifying Qualities promotes reasoning about the architecture and how well it supports palpable applications. In particular, it is important to investigate if the open architecture for building palpable systems exhibits the different palpability attributes, i.e. the palpable Qualities.

The embedded architectural prototype has been built to learn about the PalVM platform and the serial communication over IR. The testing aims at discriminating whether there are restrictions in the PalCom open architecture or if the constraints are due to the current hardware implementation (e.g IR communication implemented over serial port).

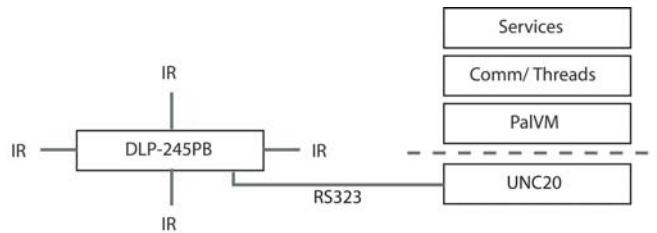


Figure 60. PalCom tile stack

The Embedded architectural prototype is the experimental prototype that allows concrete measurements of different tasks, which have been described in terms of Qualities Scenarios. The tasks are based on major performance concerns, and the Qualities Scenarios, as seen with the example above, constitute a concrete metric for measuring whether a given prototype will respond within certain meaningful ranges.

The implementation of the architecture in the hardware support poses a specific set of challenges for the empirical experiments. In fact, architectural prototypes serve to learn about specific aspects of the development, such as the technological platforms or the programming models.

Since we used the *embedded* architectural prototype, the performance testing was mainly based on the architecture running on current hardware implementation. As with most architectural prototyping, performance testing is related to timing. Events like messages, interruptions and actions from users occur throughout time and the system must respond to them. *Response time* means how long it takes to process a request, while *timeliness* means the ability to meet deadlines, i.e., to process a request in a deterministic and acceptable amount of time (O'Brien et al. 2005).

There are a number of key factors of the PalCom open architecture that are common to other service oriented architectures (SOA) that affect the performance. Primarily the open architecture deals with distributed computing. Service and devices are normally located on different machines and communication in such a scenario “takes longer than computation” (p. 79, Bass et al. 2003). The need to communicate over the network increases the response time. In our case the ad-hoc networks among the tiles provide a challenge for the near real-time application within the Active Surfaces scenario where timeliness is a core requirement.

The interaction protocol and how it is implemented affects timeliness and overall performance. The open architecture has a set of protocols (Wire, Discovery and Service Interaction protocols) running on top of each other on different layers of the architecture (PalCom Deliverable 54). Such fragmentation guarantees the flexible adoption of the interaction protocol(s). It increases the dimension of the whole interaction protocol but also allows adaptation to different implementations, especially throughout experimentation. In Active Surfaces the IR communication bandwidth challenges the use of assemblies and of the discovery protocol. Throughout profiling the communication with the Discovery protocol was estimated at 45 sec, i.e. reciprocal DeviceDiscovery and ServiceDescription and replies among the tiles. The activity analysis stands out in that, while in the programming mode the communication can take 5- 10 sec, in the playing mode it should be quicker than 2 sec.

In Active Surfaces the architecture runs on homogenous machines, thus on different platforms interoperability is not a demand. Moreover, it is crucial to understand whether vertical scalability in each device impacts the performance of the overall system of tiles and if horizontal scalability affects the response time among many communicating tiles.

The goal of the experiments with Active Surfaces is also to evaluate the improvements throughout the development of the architecture and to determine if there is an improved performance with the newly developed components. We have experimented with both the project 3rd year release and the project 4th and last year releases of the PalCom open architecture. More in details, we have made a comparison between two subsequent versions of the PalVM-debug, respectively released in autumn 2007 and at the end of the project December 2007, and the final PalVM-release. The Intermediate PalVM-debug was the version available through the developmental process. At the end of the project both the PalVM-debug and the PalVM-release versions were developed.

Throughout the performance testing we made a comparison between the two releases of the same VM version: the *Intermediate* and the *Final* PalVM-debug. Then we also made explorative testing with the PalVM-release that is the final version ready to use, also available open source with the whole PalCom architecture at the project website (www.ist-palcom.org). The PalVM debug and release versions have been solely experimented through the performance testing and not with the end-users because they have been released at the end of the project (Dec 2007), and it was too late to have it tested with the users and to be included in this thesis.

In this way it is also possible to discriminate the improvements that have been made through the development process and to better inform the interpretation of the results. In the following sections experiments with the three different PalVM versions will be documented. We followed the same experimental tasks for the two PalVM-debug versions.

Instead, for the exploration of the final PalVM-release we assumed that the Communication and Discovery tasks and the Re-configuration tasks were especially worthwhile to perform. In fact the further involve the use of IR communication and allow evaluation of architectural improvements in respect to hardware implementation, the latter allow testing “support for adaptation” that is of major interest for this application prototype. The Performance Tasks were not carried out because of resource and time restrictions.

5.3.2 Tasks

The tasks have been designed in order to translate the Qualities in measures observable via execution. The tasks aim at demonstrating how the existing architectural components would behave in performing the Active Surfaces scenario, e.g. performing the assigned activities.

The performance testing is based on a user-oriented perspective and assumes human practice in the therapeutic setting, described in Chapter 2, as baseline for the experiments. For this purpose Activity Scenarios, Envisioning, Prototype and Qualities Scenarios describe the research with Active Surfaces in very different ways, i.e. by focusing on the fieldwork, on the design and development, and on the experimentation and results.

The measurements and the results of the experiments are represented as Qualities scenarios. By using Qualities Scenarios we will report on the results and will also describe how the open architecture may support *palpability*.

In particular time responses, delays or frequency of errors have been observed with respect to the requirements coming from the activity analysis. For what regards timeliness, the major requirements from the therapeutic activity in the water are the duration of the whole session (45 minutes), the pace of the interaction (cycles of 3 to 5 minutes games to the utmost) intervened by the restless time pauses (2-3 minutes). These data allowed us to define the baseline for the experiments.

In order to determine whether there are restrictions in the PalCom open architecture or if the eventual

constraints are due to the current hardware implementation, we have organized many performance tests around two different conditions:

- Tasks in which the performance is influenced mainly by the software architecture currently running
- Tasks in which the performance is both influenced by the architecture and mostly by the current hardware implementation.

Together with Bass et al. (2003), we assume that in distributed computing communication takes longer than computation, we have designed the second situation to be mainly characterized by the use of distant communication over IR modules among the tiles.

The use of two GameServices, the *ist.palcom.tiles.test.fish.prc* and the *ist.palcom.tiles.test.timer.prc* correspondingly reify the two experimental conditions.

The Fish Puzzle game (*ist.palcom.tiles.test.fish.prc*) is the software implementation of the Jigsaw Puzzle Game described in Par. 3.14. The Fish Puzzle game consists of a fish image divided into four pieces that have to put in the right position. In this way the service involves communication on the sides and challenges the current implementation of communication modules.

The LightUp game (*ist.palcom.tiles.test.timer.prc*) is described in Prototype Scenario 7 in Par. 4.4. The LightUp game consists of several tiles that alternatively blink one after the other. The child follows the blinking lights by moving within the environment. Because a timer regulates the blinking lights in a sequence, the GameService doesn't involve any communication among the tiles.

The use of these two services (and games) allows us to compare the tasks under two different experimental conditions that suit the objectives.

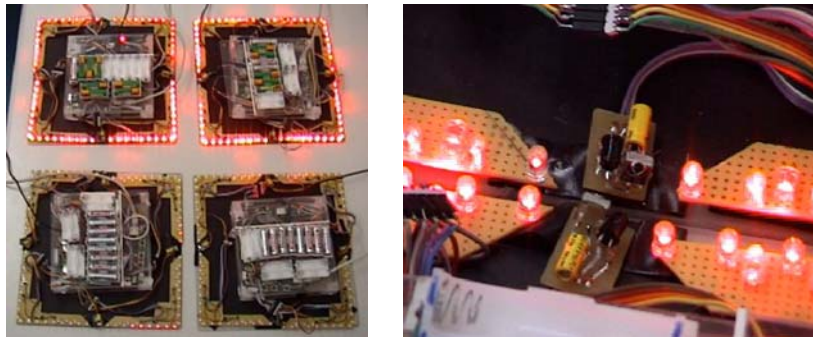


Figure 61. The experiments with the Active Surfaces embedded architecture prototype

The measurements must also be put in relation with the resources used. The experimental tasks were carried out with the four embedded architectural tiles connected via Ethernet cable to the terminal on a desktop machine.

We also isolated and neutralized those variables that could have affected the behaviour of the system, i.e. battery power consumption and possible cable disruption. Making the variables neutral allows an effective evaluation of the architectural performance. That's why throughout the testing the tiles were powered by a continuous AC current.

The telnet terminal, connected via Ethernet cable, allowed us to see and observe the performance of the PalVM running on the microprocessor. In fact telnet has been used as communication protocol between the desktop machine and the uClinux running on the embedded architectural prototype. Telnet terminal also allowed to observe the running code real time on the Console Log and to have that saved as a text file. Time measurements can thus be taken on the *Console Log*. It is possible to analyse the Log both to have the

measure of the performance of the PalVM and the hardware behaviour (i.e. the LED feedback on user actions).

Based on the existing literature (Bass et al. 2003; Bosch 2000) we arrived at the following simplified method for prototype based architectural experimentation:

1. Define goals.
2. Implement architecture prototype.
3. Execute prototype.
4. Analyse logs.
5. Discuss the results with respect to the Qualities.

While the objectives are described in the previous sections and the implementation of the architectural prototype is described in Par. 3.4.2.2, the phase (3), *Execute prototype*, consists of the proper experimentations. Executing the prototypes permits the data to be gathered for analysis at the next step. While executing the prototype we tried to match the target environment as closely as possible.

Through phase (4), *Analyse logs*, we analysed the gathered logs and extracted information regarding the Qualities that are under evaluation. The techniques defined for phase (4) utilized the Console Log Time and the Terminal windows video used a Timer software application that showed the timeline.

Throughout the tasks the architectural prototype was executed on each tile platform and the logs from all runs were gathered and stored for further analysis. The logged data was stored in separate log files, one for each tile.

We analysed the generated logs in Excel worksheet and extracted the information that we wanted from the data files. Based on the timeline and the events identifiers from the logs we were able to extract information that we had defined as necessary for evaluating the tasks, namely:

- The time it takes for the tiles to discover each other, establish communication and give feedback,
- The time it takes for the tiles to interrupt the communication and give feedback,
- The possibility to run several services in parallel,
- The possibility to use the tiles for extended amount of time.

The analysis was automated as much as possible since the amount of data easily becomes overwhelming. The tools used also provided valuable documentation for describing what happened throughout the experiments.

Discussing the results with respect to the Qualities, phase (5), means addressing the Qualities that are to be evaluated based on the information gathered from the analysed logs.

Since the experiments presented in this research are examples of exploratory testing on architectural prototypes, once all the steps have been completed and results from the analysis are available, they will be discussed with an architectural developer who could review them and give feedback for defining the future developments of the architectural prototype.

The following section describes the three categories into which the experimental tasks have been grouped: *Communication and discovery*, *Re-configuration* and *Performance*.

Communication and Discovery

Communication and Discovery comprise a set of six tasks that all involve the use of communication over IR. Communication and Discovery is a prerequisite for many palpable Qualities, from Assemblability to Experimentability. We explicitly focused on Resource Awareness, specifically regarding the PalVM ability to recognize and communicate its own 1st order resources and those owned by other neighbouring tiles.

Moreover, the current embedded architectural prototype fits with the description of and the possible experimentation with 1st order Resource Awareness.

The tasks are designed as two series each consisting of 10 repetitions of the tasks.

- In the first series the tasks are interrupted by re-boot of the game services (*Re-boot series*),
- in the other series the tasks are carried out continuously over time (*Over time series*).

The former case represent the *normal* performance the tiles have on these tasks. The latter evidences how the performance in these specific tests varies over time.

The comparison between the Re-boot and Over time series allows us to evaluate the performance of the PalVM over time, e.g whether its performance degrades or not.

Re-configuration

Re-configuration comprises 2 tasks performed under both the experimental conditions, with and without the use of communication. Thus there is a set of 4 tasks.

Re-configuration is mostly considered as a support for Adaptability, as the enabling condition for the next future development of the architecture. User and system adaptation is particularly relevant for tuning the activity based on either user decision or autonomous system behaviour.

The tiles currently can run either *fixed* GameServices, like the FishService, the DominoService, and the WordService; or *open* GameServices where the tiles are in programming mode and learn how to configure by physical programming-by-example. The tiles also run FeedbackServices, like the actual LEDService or the possible VibrationService and SoundService that can be developed in the future.

The Re-Configuration tasks can either mean:

- choosing among existing pre-defined GameServices or
- the flexible use of single services related to game configuration, e.g. tiles' sequence, sensing and feedback.

In one case the system should allow shifting between pre-defined GameServices, i.e. different games that have already been configured. In the second case the system should allow running more services at the same time

That's why we launched different services in parallel simulating the two conditions described above. We are able to compare the task under two different conditions represented by the *ist.palcom.tiles.test.fish.prc* services, which involves IR communication among the tiles; and *ist.palcom.tiles.test.timer.prc* which doesn't involve the use of IR communication.

Performance

Performance comprises 1 task performed under both the experimental conditions, with and without the use of communication. Thus there is a set of 2 tasks that consist of observing two services, the *ist.palcom.tiles.test.fish.prc* and the *ist.palcom.tiles.test.timer.prc* running for 30 min.

As mentioned above, the overall session lasts 45 minutes and the duration of a single game situation can be assumed to be 30 minutes at the very most. In fact even if it is possible that children find some games very engaging, it is very hard to carry out the same game for almost the whole session. Furthermore, as reported in Chapter 2, game dynamics usually last few minutes.

In the tables below the tasks and the procedure have been detailed.

<i>Areas</i>	<i>Tasks</i>
Communication and Discovery	<p>The tiles put together</p> <ul style="list-style-type: none"> a. 1+1 tiles: one is still, the other is rotated to reach the correct orientation b. 1+2 tiles: one is still, the other two are rotated to reach the correct orientation at the same time c. 1+3 tiles: one is still, the other three are rotated to reach the correct orientation at the same time <p>The tiles put apart</p> <ul style="list-style-type: none"> a1. 2 tiles correctly connected are kept away b1. 3 tiles correctly connected are kept away c1. 4 tiles correctly connected are kept away
Re-Configuration	<p>Running services in parallel with the <i>ist.palcom.tiles.test.fish.prc</i></p> <ul style="list-style-type: none"> d. 1+1 tiles: Launching 1+1 services e. 1+1 tiles: Launching 1+2 services <p>Running services in parallel with the <i>ist.palcom.tiles.test.timer.prc</i></p> <ul style="list-style-type: none"> f. 1+1 tiles: Launching 1+1 services g. 1+1 tiles: Launching 1+2 services
Performance	<p>Running the following services over 30 min.</p> <ul style="list-style-type: none"> h. Performance with the <i>ist.palcom.tiles.test.fish.prc</i> i. Performance with the <i>ist.palcom.tiles.test.timer.prc</i>

Table 62. Experimental tasks carried out with the Embedded architectural prototype.

<i>Tasks</i>		<i>Procedures</i>	
Communication and Discovery	Task (a) Task (b) Task (c)	<i>Actions</i> Open connection	<i>Data</i> First (malformed) data packet First Discovery message
	Task (a1) Task (b1) Task (c1)	Close connection	Last Discovery message Last CLEAR message*
		Re-boot series - (10 times) Over time series - (10 times)	
Re-Configuration	Task (d) Task (e) Task (f) Task (g)	Launching the main GameService (<i>ist.palcom.tiles.test.fish.prc</i> and <i>ist.palcom.tiles.test.timer.prc</i>) and other services in parallel: - <i>ist.palcom.tiles.test.led.prc</i> - <i>ist.palcom.tiles.test.tetris.prc</i> - <i>ist.palcom.tiles.test.IR.prc</i>	
Performance	Task (h) Task (i)	Launch the main GameServices and observe the performance over time (30min): - <i>ist.palcom.tiles.test.led.prc</i> - <i>ist.palcom.tiles.test.tetris.prc</i>	

Table 63. Experimental tasks and procedures

5.3.3 Results

5.3.3.1 Intermediate PalVM-debug version

Communication and discovery

Task (a)

Two tiles are put together: 1+1 tiles, one is still, the other is rotated to reach the correct orientation for the side connection.

In the Re-boot series the GameService *ist.palcom.tiles.test.simplefish.prc* is launched for every discovery action.

9.7 sec is the average time for the OpenConnection action/First Discovery message received in the Re-boot series (max = 79 sec, min = 2 sec).

13.55 sec is the average time for the OpenConnection action/First discovery message received in the Over time series. The response time doesn't vary (i.e. increase) over time. It has a higher variation (max = 60 sec, min = 2 sec) with respect to the Re-boot series.

Task (a1)

Two correctly connected tiles are kept apart.

11.55 sec is the average time for the CloseConnection action/last CLEAR message (max = 23 sec, min = 0 sec). It has to be considered in conjunction with the Last Discovery message/ CLEAR message index: 18.65 sec is the average in the Re-boot series.

* CLEAR Message is the output for the *print.out* command related to LEDs feedback turn off.

14.1 sec is the average time for the CloseConnection action/CLEAR message (max = 40 sec, min = 0 sec) in the Over time series.

19.3 sec is the average time for the Last Discovery message/CLEAR message in the Over time series. The difference in time response between Close connection action and the first CLEAR messages circulation gives a measure of about two tiles put aside and the response time to turn off the LEDs.

Task (b)

Three tiles are put together: 1+2 tiles, one is still, the other two are rotated to reach the correct orientation at the same time.

19.26 sec is the average time for the OpenConnection action/First Discovery message in the Re-boot series. Tile (3) is connected to both of the other two tiles and doesn't show a performance that is worse than the others. In fact, it has an average response time of 7.2 sec in comparison to the 40 sec and 9.1 sec average response time for tile (4) and tile (6).

15.8 sec is the average for OpenConnection action/ First Discovery message (max = 102 sec, min = 2 sec) in the Over time series.

Task (b1)

Three correctly connected tiles are kept apart

13.44 sec is the average time for the CloseConnection action/CLEAR message in the Re-boot series.

18.9 sec is the average time for the Last Discovery message/CLEAR message in the Re-boot series.

17.2 sec is the average time for the CloseConnection action/CLEAR message in the Over time series.

19.3 sec is the average time for the Last Discovery message/CLEAR message in the Over time series.

Task (c)

Four tiles are put together: 1+3 tiles, one is still, the other three are rotated to reach the correct orientation at the same time.

Tile 3 (6) was connected to all the three tiles and has 3 connections to manage.

12.6 sec is the average time for the OpenConnection action/First Discovery message in the Re-boot series.

16.2 sec is the average time for the OpenConnection action/ First Discovery message (max = 134 sec, min= 2 sec) in the Over time series.

Task (c1)

Four correctly connected tiles are kept apart.

12.9 sec is the average time for the CloseConnection action/CLEAR message in the Re-boot series.

12.5% of all the CLEAR messages arrived before the CloseConnection action (up to 30 sec before). In these cases there hasn't been any delay that can affect the average time response reported in CloseConnection action/CLEAR message. This data affects the actual use scenarios: while the tiles are still connected, LEDs light feedback has already turned off.

20.4 sec is the average time for the Last Discovery message/CLEAR message in the Re-boot series.

20 sec is the average time for the CloseConnection action/CLEAR message (max=92 sec, min= 0 sec) in the Over time series.

Tile (3) was connected to all of the other three tiles and had to manage a bigger amount of connection data. Tile 3 performance varied over time in respect to Task (c1).

The increase of response time throughout the ten repetitions is:

Tile 3 (19, 6, 13, 32, 35, 54, 35, 43, 92, 84)

The other tiles have one side connection and their progression through the test is shown below:

Tile 1 (22, 0, 12, 13, 9, /, 15, 12, 8, /)

Tile 2 (23, 17, 22, 21, 21, 21, 19, 22, 21, 21)

Tile 4 (12, 13, 9, 6, 7, 7, 18, /, 3, 14)

The “ / “ indicates the case when the CLEAR message arrived before the close connection action (75 sec, 20 sec and 1 sec before), in these cases there hasn't been any delay that can affect the average time response reported in the CloseConnection action/CLEAR message.

21 sec is the average time for the Last Discovery message/CLEAR message in the Over time series.

In the tables below we have summarized the results of the performance testing with the Intermediate PalVM-debug version.

	<i>2 Tiles</i>	<i>3 Tiles</i>	<i>4 Tiles</i>
Re-boot series	9.7	19.2	12.6
Over time series	13.5	15.8	16.2

Table 64. Put together action - Intermediate PalVM-debug version

	<i>2 Tiles</i>	<i>3 Tiles</i>	<i>4 Tiles</i>
Re-boot series	11.5	13.44	19.3*
Over time series	14	17.2	20

Table 65. Keep away action - Intermediate PalVM-debug version

Quality Scenario – (Support for) Resource Awareness

The therapist is trying to inspect the communication and discovery between the tiles. She puts together two tiles and they light up on the correct sides after 13.55 sec. She then holds the tiles apart and they take 23 sec to turn the light feedback off.

Re-configuration

Task (d)

The *ist.palcom.tiles.test.fish.prc* was launched in parallel with the *ist.palcom.tiles.test.led.prc* service. Two services running simultaneously are well supported and the hardware (IR) works well. The LEDs strips are alternatively used by each service as shown below (from Tile4 Re-configuration-Task (d) Log):

```
[...]  
LED CONFIGURATION 1  
LED CONFIGURATION 2  
IM TILE-4-ON MY PORT-2-IM CONNECTED TO TILE-3-PORT-2-END  
HAPPY:0200  
LED CONFIGURATION 1  
LED CONFIGURATION 2
```

* 12.50% of malfunctions, i.e. the CLEAR messages are received before the CloseConnection action.

```
IM TILE-4-ON MY PORT-2-IM CONNECTED TO TILE-3-PORT-2-END
HAPPY:0200
<> Checksum error. Discarding!
LED CONFIGURATION 1
LED CONFIGURATION 2
[...]
```

Task (e)

The *ist.palcom.tiles.test.fish.prc* was launched in parallel with the *ist.palcom.tiles.test.led.prc* and the *ist.palcom.tiles.test.tetris.prc* service. Three services running simultaneously are supported, but the behaviour of the system is affected. Regarding the Puzzle Fish game the communication is delayed in comparison to the normal condition, where only the Fish GameService runs. The tiles receive incoming SideHappy messages on average every 2 min after the open-connection action takes place.

Task (f)

The *ist.palcom.tiles.test.timer.prc* was launched in parallel with the *ist.palcom.tiles.test.led.prc* service. Two services running simultaneously are well supported by the current implementation. The LEDs strips are alternatively used by each service and, when concurrent actions take place that ask for LEDs actions, they are scheduled in a contingent and efficient way.

Task (g)

The *ist.palcom.tiles.test.timer.prc* was launched in parallel with the *ist.palcom.tiles.test.led.prc* and the *ist.palcom.tiles.test.tetris.prc* service. These three services running simultaneously are well supported.

The *ist.palcom.tiles.test.timer.prc* was also launched in parallel with the *ist.palcom.tiles.test.led.prc* and the *ist.palcom.tiles.test.ir.prc* service. The latter is specifically an IR test communication service. There wasn't support for this combination of services because the amount of resources required by this service exceeded the memory currently allocated.

This is documented in the following excerpt of the Log:

```
# ./pal-vm-unc20-debug -cp . ist.palcom.tiles.test.led.prc &
57
# ./pal-vm-unc20-debug -cp . ist.palcom.tiles.test.timer.prc &
58
# ./pal-vm-unc20-debug -cp . ist.palcom.tiles.test.ir.prc
# Warning, new block allocation gave a too small block - system out of memory?
LED CONFIGURATION 1
LED CONFIGURATION 2
LED CONFIGURATION 1
[...]
```

Quality Scenario – (Support for) Adaptability

The tiles system initiates a newly created game composed by the LED Feedback Service, the Vibrate Feedback service and the Tetris Game Service. The tiles, when correctly combined, take an average of 2 min to correctly light up and vibrate.

Performance

Task (h)

The Puzzle Fish Game has been evaluated in action over time. The test has been organized in order to match the interactivity pattern exhibited by the LightUp GameService. In doing this, a series of

10 connection slots have been carried out over 30 minutes. Each slot had an average duration of 2 min with an average 1 min break among the slots.

The performance of the overall system doesn't decrease over time. It uniformly maintains the same behavioural pattern.

The test proved that malfunctions uniformly occurred through the 10 communication slots. There have been 18 malfunction episodes in 40 global test episodes (10 comm. Slots x 4 tiles). Throughout these episodes, during the 2 min open-connection slot, one of the two IR side connections received an average of just 2 incoming messages while the other received incoming messages in an almost continuous way. In one episode the system completely missed one side connection along the open-connection slot.

After every close-connection action it takes an average of 10 sec before every tile has received the final CLEAR message that turns off the connection LEDs. In two cases, at the beginning of the test, the CLEAR message took in one case 40 sec and in the other more than 50 sec to circulate.

Task (i)

The LightUp GameService has been evaluated in action over time. The LightUp game doesn't involve the use of IR communication. Throughout Task (i) the performance of the overall system was good and all the tiles behaved as expected.

Quality scenario – (Support for) Experimentability

The child puts tile_04 in the correct position between tile_06 and tile_07 in the Fish Puzzle Game and, in half the cases, while the 04-07 connection immediately lights up the 04-06 connection lights up after more than a minute for an average of 10 sec.

5.3.3.2 Final PaIVM-debug version

Communication and discovery

Task (a)

Two tiles are put together: 1+1 tiles, one is still, the other is rotated to reach the correct orientation for the side connection.

In the Re-boot series the GameService *ist.palcom.tiles.test.simplefish.prc* is launched for every discovery action.

5.25 sec is the average time for the OpenConnection action/First Discovery message received in the Re-boot series (max = 40 sec, min = 1 sec).

5.05 sec is the average time for the OpenConnection action/First discovery message received in the Over time series (max = 16 sec, min = 1 sec). The response time doesn't vary (i.e. increase) over time.

Task (a1)

Two correctly connected tiles are kept apart.

11 sec is the average time for the CloseConnection action/last CLEAR message (max = 15 sec, min = 3 sec) in the Re-boot series.

11.4 sec is the average time for the CloseConnection action/CLEAR message (max = 40 sec, min = 0 sec) in the Over time series.

Task (b)

Three tiles are put together: 1+2 tiles, one is still, the other two are rotated to reach the correct orientation at the same time.

6.5 sec is the average time for the OpenConnection action/First Discovery message in the Re-boot series (max = 29 sec, min = 1 sec). Tile (3) is connected to both of the other two tiles and doesn't show a performance that is worse than the others.

8.7 sec is the average for OpenConnection action/ First Discovery message (max = 70 sec, min = 1 sec) in the Over time series.

Task (b1)

Three correctly connected tiles are kept apart

13 sec is the average time for the CloseConnection action/CLEAR message in the Re-boot series (max = 16 sec, min= 5 sec).

11.46 sec is the average time for the CloseConnection action/CLEAR message in the Over time series (max = 16 sec, min= 2 sec).

Task (c)

Four tiles are put together: 1+3 tiles, one is still, the other three are rotated to reach the correct orientation at the same time.

Tile 3 (6) was connected to all the three tiles and has 3 connections to manage.

6.8 sec is the average time for the OpenConnection action/First Discovery message in the Re-boot series (max = 51 sec, min= 1 sec). It is worth to note that tile 6, which was the only one connected on three sides, exhibited better performance than the other tiles.

6.95 sec is the average time for the OpenConnection action/ First Discovery message (max = 28 sec, min= 1 sec) in the Over time series. Since 20 min henceforth one tile was overloaded by too many incoming malformed data packet and this has delayed the incoming communication messages.

Task (c1)

Four correctly connected tiles are kept apart.

14.55 sec is the average time for the CloseConnection action/CLEAR message in the Re-boot series (max = 34 sec, min= 7 sec).

13.7 sec is the average time for the CloseConnection action/CLEAR message (max = 70 sec, min = 0 sec) in the Over time series.

15% of all the CLEAR messages arrived before the CloseConnection action. In these cases there hasn't been any delay that can affect the average time response reported in CloseConnection action/CLEAR message.

In the tables below we have summarized the results of the performance testing with the Final PalVM-debug version.

	2 Tiles	3 Tiles	4 Tiles
Re-boot series	5.25	6.5	6.8
Over time series	5.05	8.7	6.95

Table 66. Put together action - Final PalVM-debug version

	2 Tiles	3 Tiles	4 Tiles
Re-boot series	11	13	14.55
Over time series	11.4	11.46	13.7 [†]

Table 67. Keep away action - Final PalVM-debug version

Quality Scenario – (Support for) Resource Awareness

The therapist is trying to inspect the communication and discovery between the tiles. She puts together two tiles and they light up on the correct sides after 5 sec. She then holds the tiles apart and they take 11 sec to turn the light feedback off.

Re-configuration

Task (d)

The *ist.palcom.tiles.test.fish.prc* was launched in parallel with the *ist.palcom.tiles.test.led.prc* service. Two services running simultaneously are well supported and the hardware (IR) works well.

Task (e)

The *ist.palcom.tiles.test.fish.prc* was launched in parallel with the *ist.palcom.tiles.test.led.prc* and the *ist.palcom.tiles.test.tetris.prc* service. These three services running simultaneously are badly supported. Regarding the Puzzle Fish game the communication is delayed in comparison to the normal condition. The tiles receive incoming SideHappy messages on average every 3 min after the open-connection action takes place and took about 20 sec for circulating CLEAR message after the CloseConnection action.

Task (f)

The *ist.palcom.tiles.test.timer.prc* was launched in parallel with the *ist.palcom.tiles.test.led.prc* service. Two services running simultaneously are well supported by the current implementation.

Task (g)

The *ist.palcom.tiles.test.timer.prc* was launched in parallel with the *ist.palcom.tiles.test.led.prc* and

[†] 15% of malfunctions, i.e. the CLEAR messages arrived before the CloseConnection action

the *ist.palcom.tiles.test.tetris.prc* service. These three services running simultaneously are well supported. The tiles receive incoming SideHappy messages on average every 5 seconds after the open-connection action takes place.

Performance

Task (h)

The Puzzle Fish Game has been evaluated in action over time. Through the 10 connection slots (2 min each) 6.5 sec is the average time for the OpenConnection action/First Discovery message (max = 48 sec, min= 1 sec). The performance of the overall system doesn't decrease over time. It uniformly maintains the same behavioural pattern.

After every close-connection action it takes an average of 21 sec before every tile has received the final CLEAR message that turns off the connection LEDs. Exceptionally, one tile hasn't exhibited CLEAR messages at all. It means that it didn't turn the LEDs off through the whole task.

Task (i)

The LightUp GameService has been evaluated in action over time. The LightUp game doesn't involve the use of IR communication. Throughout Task (i) the performance of the overall system was good and all the tiles behaved as expected.

Quality scenario – (Support for) Experimentability

The child puts tile_04 in the correct position between tile_06 and tile_07 in the Fish Puzzle Game and the tile connections light up after 6.5 sec.

5.3.3.3 Final PaIVM-release exploration

Communication and discovery

Task (a)

Two tiles are put together: 1+1 tiles, one is still, the other is rotated to reach the correct orientation for the side connection.

3.2 sec is the average time for the OpenConnection action/First Discovery message received in the Re-Boot series (max = 8 sec, min = 1 sec).

3.5 sec is the average time for the OpenConnection action/First Discovery message received in the Over time series (max = 9 sec, min = 1 sec).

Task (a1)

Two correctly connected tiles are kept apart.

7.6 sec is the average time for the CloseConnection action/last CLEAR message in the Re-Boot series.

8.1 sec is the average time for the CloseConnection action/last CLEAR message in the Over time series.

Task (b)

Three tiles are put together: 1+2 tiles, one is still, the other two are rotated to reach the correct orientation at the same time.

7 sec is the average time for the OpenConnection action/First Discovery message in the Re-Boot series (max = 32 sec, min = 1 sec).

7.6 sec is the average time for the OpenConnection action/First Discovery message in the Over time series (max = 50 sec, min = 1 sec).

Task (b1)

Three correctly connected tiles are kept apart

9.8 sec is the average time for the CloseConnection action/CLEAR message in the Re-Boot series (max = 34 sec, min = 0 sec).

10.6 sec is the average time for the CloseConnection action/CLEAR message in the Over time series (max = 20 sec, min = 0 sec).

Task (c)

Four tiles are put together: 1+3 tiles, one is still, the other three are rotated to reach the correct orientation at the same time.

6.9 sec is the average time for the OpenConnection action/First Discovery message in the Re-Boot series (max = 32 sec, min = 1 sec).

7.5 sec is the average time for the OpenConnection action/First Discovery message in the Over time series (max = 40 sec, min = 1 sec).

Task (c1)

Four correctly connected tiles are kept apart

12.5 sec is the average time for the CloseConnection action/CLEAR message in the Re-Boot series (max = 44 sec, min = 0 sec).

13.3 sec is the average time for the CloseConnection action/CLEAR message in the Over time series (max = 50 sec, min = 1 sec).

In the tables below we have summarized the results of the performance testing with the Final PalVM-release.

	<i>2 Tiles</i>	<i>3 Tiles</i>	<i>4 Tiles</i>
Re-boot series	3.2	7	6.9
Over time series	3.5	7.6	7.5

Table 68. Put together action - Final PalVM- release

	<i>2 Tiles</i>	<i>3 Tiles</i>	<i>4 Tiles</i>
Re-boot series	7.6	9.8	12.5
Over time series	8.1	10.6	13.3

Table 69. Keep away action - Final PalVM- release

Re-configuration

Task (d)

The *ist.palcom.tiles.test.fish.prc* was launched in parallel with the *ist.palcom.tiles.test.led.prc* service. Two services running simultaneously are well supported and the hardware (IR) works well.

Task (e)

The *ist.palcom.tiles.test.fish.prc* was launched in parallel with the *ist.palcom.tiles.test.led.prc* and the *ist.palcom.tiles.test.tetris.prc* service. These three services running simultaneously are well supported. The tiles receive incoming SideHappy messages 5 sec (average time) after the open-connection action takes place and took about 15 sec for circulating CLEAR message after the CloseConnection action.

Task (f)

The *ist.palcom.tiles.test.timer.prc* was launched in parallel with the *ist.palcom.tiles.test.led.prc* service. Two services running simultaneously are well supported.

Task (g)

The *ist.palcom.tiles.test.timer.prc* was launched in parallel with the *ist.palcom.tiles.test.led.prc* and the *ist.palcom.tiles.test.tetris.prc* service. These three services running simultaneously are well supported. The tiles receive incoming SideHappy messages on average every 5 seconds after the open-connection action takes place, defined by the Tetris game.

5.3.4 Discussion

Based on the results presented in the previous paragraph we may discuss the findings of the performance testing experiments. The discussion of the results is related to the activities addressed in the Active Surfaces scenario, with a special focus on the Use Practices described in the Activity Scenarios (see Par. 2.4).

In the tables below we have summarized the results of the performance testing with the Intermediate and the Final PalVM-debug versions and the PalVM-release version related with the *Communication and Discovery*. Data regarding the to PalVM-debug versions we compared are highlighted in grey.

The results are presented regarding the two series of gathered data (Re-boot and Over Time series), the two main actions (Put Together and Put Apart) and the scalability factor represented by the number of tiles utilized (2, 3 or 4 tiles).

Re-boot – Put together

	<i>2 Tiles</i>	<i>3 Tiles</i>	<i>4 Tiles</i>
Intermediate VM-debug	9.7	19.2	12.6
Final VM-debug	5.25	6.5	6.8
VM release	3.2	7	6.9

Re-boot – Put apart

	<i>2 Tiles</i>	<i>3 Tiles</i>	<i>4 Tiles</i>
Intermediate VM-debug	11.5	13.44	19.3*
Final VM-debug	11	13	14.55
VM release	7.6	9.8	12.5

Over time – Put together

	<i>2 Tiles</i>	<i>3 Tiles</i>	<i>4 Tiles</i>
Intermediate VM-debug	13.5	15.8	16.2
Final VM-debug	5.05	8.7	6.95
VM release	3.5	7.6	7.5

Over time – Put apart

	<i>2 Tiles</i>	<i>3 Tiles</i>	<i>4 Tiles</i>
Intermediate VM-debug	14	17.2	20
Final VM-debug	11.4	11.46	13.7‡
VM release	8.1	10.6	13.3

We start by considering the communication between two tiles. The results show that the discovery between two tiles that have been put together in the correct way can take from 2 sec to more than a minute (79 sec.) with the Intermediate PalVM-debug. Comparing the average of 9.7 sec of this version with the average of

* 12.50% of malfunctions, i.e. the CLEAR messages are received before the CloseConnection action.

‡ 15% of malfunctions, i.e. the CLEAR messages arrived before the CloseConnection action.

5.25 obtained with the Final PalVM-debug version, we conclude that considerable improvement has been made toward the requirements coming from the activity analysis.

The ‘Over Time’ series shows the performance of a possible real scenario where the tiles are continuously used throughout time. In a real setting such as this it shouldn’t be desirable to re-boot the system every time to get the best performance as possible. The performance under these conditions is worst than the average Re-boot condition in fact response times are higher than in the Re-boot series. We can also conclude that the overall performance of the system doesn’t decrease significantly throughout time in the Re-boot or in the ‘Over Time’ series.

The results on communication between two tiles with the Intermediate PalVM-debug (average 13.5 sec) were far from what was required from the scenario while they were considerably better with the Final PalVM-debug version (average 5.05 sec). The improvement is even more established by the average of 3.5 seconds obtained with the Final PalVM-release. These late results prove to be close to what expected from user perspective, i.e. what we know from the activity analysis (2 sec).

Regarding two tiles that are kept apart, the average values with both the PalVM-debug versions for receiving the CLEAR message in the two conditions (11.5 sec/ 14 sec; 11 sec/ 11.4 sec) are far from the desired performance. With the final PalVM-release there is improvement (average of 8.1 sec) but it is still not good enough. In fact in order to be coherent, the feedback of the tiles should occur within 2 sec after the input action.

Such a contingent reaction of the system would fulfil the requirement of providing the children with an engaging and motivating interaction. In fact, if it occurs during the programming, the communication and broadcast among the tiles and between them and the Assembler Tile can take even 5- 10 sec; during play it should be quicker than 2 sec.

The data regarding the CloseConnection action have to be considered in conjunction with the values related to the Last Discovery message and the CLEAR messages. In fact, independently from when the user may perform the CloseConnection actions, there is latency even before the CLEAR messages are circulated, which is determined by the last Discovery message received. With the Intermediate PalVM-debug version the delay increases as the number of connections increases, as it is shown in the table below.

	<i>2 Tiles</i>	<i>3 Tiles</i>	<i>4 Tiles</i>
Re-boot series	19.3	18.9	20.4
Over time series	18.65	19.3	21.2

Table 70. Delay between the Last Discovery message and the feedback, i.e. LEDs off (CLEAR message) - Intermediate PalVM-debug version

With the Final PalVM-debug version the delay is reduced until an average time of 15 sec with 2, 3 and 4 tiles and, with the PalVM-release it is further lowered to 10 sec.

The interpretation of the results regarding the CloseConnection action/CLEAR messages is thus interwoven with the latency represented by the Last Discovery message/CLEAR messages. In fact the user CloseConnection action always occurs within the latency period of time. When it occurs at the end of the latency, the response time of the tiles is very short and the two events might seem to be related. Notwithstanding this factor, the data are very informative in this research since they describe how the use of the system is coupled with its functioning.

The comparison among the three conditions, represented by the communication and discovery between 2 Tiles, among 3 Tiles and 4 Tiles, and the improvements made between the intermediate and the final VM versions also gives a quantitative measure of how horizontal scalability affects the performance of the tiles system.

Active Surfaces is conceived and designed as a modular system that is currently implemented in four units. In future implementation we expect to produce a higher number of tiles, up to 12 units. The experimental data suggest that the performance of PalVM and the PalCom Communication components should be improved to meet the requirements of a highly scalable system and guarantee acceptable time responses as the number of the modules increase.

In this perspective the improvements made with the Final VM-debug and VM-release show good support for scalability. In fact the performance of 3-tiles and 4-tiles communication is almost analogous being an average of 6.5 sec for 3-tiles open communication and 6.8 sec for 4 tiles, and 8.7 sec and 6.95 sec respectively for 3-tiles and 4-tiles close communication. This may be considered a promising starting point for the future PalVM implementation.

Regarding *Re-configuration*, the tasks aim at informing the support needed for future development of Adaptability. This is done by showing the potentials of the PalVM in supporting several services running in parallel and thus also creative combinations and adaptations of the tiles system.

The results of Task (d); (e); (f); (g) contribute to clarify whether it is possible to choose existing GameServices and shift among these pre-defined solutions or flexibly combine single services related to game configuration (e.g. game logics, sensing and feedback).

The eventual shifting among GameServices would be affected by the time required by new services to start, about 20 sec with the Intermediate PalVM-debug and 10 sec with the final VM-release. In fact, the pace of the activity in the Active Surfaces scenario (see Par. 2.3.2.1) would impose a quicker response time for the re-configuration of the system. It is estimated to be no more than 10 sec in order to really provide the user with the experience of ready-at-hand tools.

The results described in the paragraphs above show that there is limited support for the multiple services combination with the Intermediate PalVM-debug version and that there are improvements with the Final PalVM versions.

Regarding the combination of services, all the VM versions well support two services running in tandem both in tasks involving the use of IR communication or not. Simultaneously running two services, the system coherently exhibits the behaviours defined by the two services. Three services running in parallel are also supported but it seems to affect the behaviour of the tiles by decreasing the overall performance with both the intermediate and final VM-debug versions. There is a promising improvement with the PalVM-release, the results are close to what happen with running one service alone and this could prove valuable support for re-configuration. Lighter services and more computational power are thus required in order to provide effective support for Adaptability in practice.

Tasks (i; h) regard the specific *Performance* of two GameServices, the Fish Puzzle Game and the LightUp over long periods of time. The comparison between the two results allows us to conclude that, especially with the Intermediate PalVM-debug version, the current implementation restricts the overall performance of the tiles. In fact, the performance through the LightUp GameService proved to be optimal, while task (i), which involved the communication modules, resulted in a series of malfunctions that negatively affected the overall performance. Considerable improvements are made with the final VM-debug version, which proved to be more robust and quick than the intermediate one.

5.4 Objective 2 _ User testing

5.4.1 Methods

We explore the use of Active Surfaces in real settings (objective 2) throughout a series of User testing. The field exploration of the Lightweight (Par. 4.3.1.2) and the Architectural prototypes (Par. 4.3.2) aimed at collecting requirements for architectural development by giving users the opportunity to test Active Surfaces in the setting of the swimming pool. We observed emergent uses of the prototypes and evaluated different aspects of the activity, particularly regarding games, motivation and engagement supported by Active Surfaces.

We also collected indications about the design of the prototypes, which could inform the future development of the application. The ultimate goal of this activity is to see how Active Surfaces could impact key Use Practices and provide a further conceptual exploration of the Qualities for palpability.

The user testing consisted of a range of empirical studies of Active Surfaces prototypes in different therapeutic and play settings. The testing involved the active participation of key actors, including patients, therapists, trainers and others who are involved with the care of children with special needs. The empirical and participatory focus informed testing and evaluation throughout the experimentations. It played a critical part in the analysis and assessment of prototypes, and underpinned the iterative development of the PalCom software architecture.

Throughout the user testing we assessed the design concept and the prototypes we have been developing throughout the process by experimenting with the actual skills, understanding and practices of the users. In the experimentation with the Active Surfaces we focused on how the system impacts the current practices and observed if users were empowered to do something novel which was not possible before in the absence of these tools. That's why, together with the evaluation of task efficiency, we also observed whether a task was made to be more interesting, motivating, or engaging for the users.

In the analysis of the user experience we tried to not force a situation to occur, in order to avoid the possibility that the results would not reflect the actual usage. Alternatively the users were allowed to freely explore and use the system as much as they liked. We observed people using Active Surfaces in their everyday conduct: this was really critical to the identification and the elicitation of the requirements that the application must have in order to be *palpable* for the users.

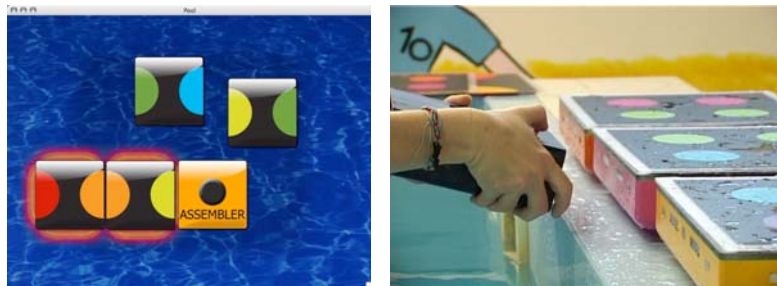


Figure 71. The Active Surfaces simulated architecture prototype (left) and the embedded lightweight prototype (right)

The Simulated prototype was used with the therapists in participatory sessions outside the water while the Lightweight embedded prototype was used with children with special needs in real settings (i.e the swimming pool).

The prototypes we adopted represent different instances of the system. In the case of the Lightweight non-architectural prototype, it was possible for the users to experience the system as it was originally conceptualized. By adhering to real conditions, such as power constraints, weight constraints and material issues, the experiments with this prototype revealed to be very useful in the creation of the final system. They also give valuable insight on different usability issues part of our investigation of *practices* and *architectural qualities*.

The Tiles Simulator, rather, provided the opportunity to test the potentials of the architecture within the simulated application of Active Surfaces running on a desktop machine. In contrast to the other prototype, the users were in a simulated environment where all the constraints of the physical environment were not present. The Simulator represented a protected and simplified environment wherein aspects of the usability can be tested in a different ways. In particular the experiment with the simulated tiles gathered data related to the PalCom assemblies and to how they might be rendered easy-to-use and understandable for the users.

The Architectural and Lightweight prototypes provided respectively logical-functional and use-oriented requirements for developing and using palpable assemblies. The assemblies are analysed from the user's perspective in order to clarify the role they may play and which applications are expected. Their use requires intuitive interaction skills; basically physical construction and sequence composition. In interacting with the tiles, both therapists and patients enact physical interaction modalities, the former through physical programming by example, the latter by manipulating, moving and combining them. The *use* of the tiles also regards elements of interactivity such as recognizing whether or not they are correctly connected, the IDs and the game logics. Moreover, the assemblies have to remain opaque in the sense that they have to be efficient tools for users who are pursuing their ultimate goals.

During the user testing we used ethnographic methods for data collection. In particular, the method we adopted was based on video recording. In *video-based field studies*, field observation and video recording are combined together. This allowed for an attentive examination of the practices and interactions that were video-recorded in the various settings. Videos provide opportunities for recording activities and events and subjecting them to detailed analysis. They also allow examining the details of the activities, by providing resources in order to undertake fine-grained investigations.

With this method we aimed at documenting and analysing the field trials of the prototype technologies. By analysing videos we analysed naturally occurring social action and interaction. Video recordings in this research have the advantage of providing access to the richness and complexity of the interactions with Ubiquitous Computing systems to be subjected to detailed analysis.

Through this exploratory study we empirically reported on samples from the observed activity. We paid attention to the resources utilized and the practical knowledge employed by users while they carried out their activities. We were interested in the communication and bodily conduct of both the therapist and the disabled child and how they were interwoven and sequentially or causally organised. The discussions with the therapists after the testing sessions provided us with a further validation of our results as well as valuable comments and suggestions.

The outcomes of the exploratory trials derived from the observation and the follow-up discussions with the therapists. Therefore, user-testing results are not quantitative data; rather, they are qualitative observations of the trials that provide valuable insights regarding the objectives of the experiments. We summarized each outcome in the results described in the following paragraphs.

These methods allowed us to analyse the impact Active Surfaces may have and the emergence of *palpability* in use. Palpability was explored through its attributes, i.e. the Qualities addressed in this research. In particular, the User testing in the swimming pool with the Lightweight prototype provided information related to:

- construction play (Assemblability),
- re-configuration and physical programming-by-example (User adaptation),
- debugging (1st Order Resource awareness)
- general exploration of the system (Experimentability).

The User testing with the Simulated prototype provided information related to:

- playing with assemblies (Assemblability),
- creation and dynamic change of assemblies (User adaptation),
- game management (2nd Order Resource awareness)
- general exploration of the environment (Experimentability).

5.4.2 Exploratory trials

5.4.2.1 Lightweight embedded prototype in the water therapy

Throughout the whole design process we have experimented with the Lightweight embedded prototype (Par. 3.4.1.2) in different situations. We have also used different releases of the prototype for both technology and design evaluations with users, see also Par. 3.4.1. In particular, we have carried out field explorations at the swimming pool in Siena with the Disabled Children Parents' Association and at the D. Chiossone rehabilitation centre in Genova. For a more detailed description see Par. 2.3.

The exploratory trials we have conducted with the users are described in the following section.

1_ Prototype exploration (January 2007)

This trial was the first time that the prototype was tested in the water with disabled people. In the first part of the test the researchers tried out the tiles by moving them around in the swimming pool, testing the connections and determining if they floated in a safe way. Since it gave positive responses, throughout the second part of the trial we involved disabled people in the use of the prototype. Both an adult with multiple disabilities and a child with moderate mental delay were allowed to hold the tiles and try them in the water. They explored the tiles and moved, rotated and pushed them in order to see what they did. The test took 50 minutes and involved one researcher and two disabled individuals assisted by one trainer.



Figure 72. On the left. LightUp Game On the right. Group activities (Session 4).

2_ Early exploration of the Fish Puzzle Game (February 2007)

The trial involved two disabled boys in an early exploration of one of the first games realized, the Fish Puzzle. One user had physical impairments (i.e. difficulty to walk autonomously) and a slight mental delay. The other had severe physical impairments and was able to move autonomously only in the water. They used the four tiles to create the fish with the aid of a special education trainer. The researcher provided support with regard to the configuration and the use of the system. The tests took an average of 30 min each. In this release the prototype used one power LED to provide feedback.

3_ Refined prototype exploration (March 2007)

The trial was mainly oriented to test the last release of the prototype. The code was optimized and the hardware was improved especially regarding the LEDs feedback (i.e. from one to five powerLEDs). We made two users try the prototype out. One of them had already tried it through test (2) and gave reliable feedback on the use. We had the opportunity to see whether the interaction was meaningful for the users and the visual feedback consistent with their actions. The whole test took 50 min.

4_ Exploration of social games and group activities (June 2007)

After the previous explorative tests, in this phase we conducted two trials with the intention of reproducing an actual play session in the water. We were interested in letting the trainer and the children use the tiles as they preferred. There were few pre-defined game configurations but they were also invited to freely make use of the prototype. We involved both children with special needs and others with normal profiles in group activities, four children for each session. In the first case there was a child with an Autistic Spectrum disorder playing with peers. In the second, a group of very young children (4-5 years old) played together. A special education trainer guided the sessions and engaged the children with the Fish Puzzle, the Domino and the Crossword games. Each session lasted 1h.

5_ Exploration of Active Surfaces in the therapeutic intervention in the water (November 2007)

This trial consisted of two sessions in which the prototype was used in the therapeutic setting with two disabled children. In this trial the current prototype implementation was experimented with for the first time. The users could try the four tiles plus the Assembler Tile and configure the games with a variety of Game Surfaces. Two therapists working for approximately 1 hour with their patients in the water used the prototype within their work practice. They tried to explore the system in light of the therapeutic objectives they were currently pursuing through the therapy.

In the first session (Session A) a 7-year-old girl with mild cognitive delay and physical impairments was involved. The therapist had been treating her both for attention and object manipulation tasks for more than a year.

In the second session (Session B) an 8-year-old boy with developmental delay was involved. He had started water therapy for one month and he still had to familiarize himself with the environment and the therapist. The researcher trained the two therapists for one hour and showed them the prototype and the potentials it might offer. They were then free to use it as they preferred in order to engage the children, maintain their practices and pursue their objectives.



Figure 73. On the left. End-user testing (Session A). On the middle and on the right. Group activities (Session 4). As described above, the exploratory field trials we conducted with the users had very different foci and provided qualitatively different outputs. In some cases (1 and 3) we mainly wanted to test and evaluate the prototype and the developments made within the actual context of use. In other cases we were definitely more interested in how trainers and children could use the prototype (2 and 4) and what their reactions and the emerging interactions were. In another case (5) we wanted to give the water therapists a system that was as robust and usable as possible to try out during the therapeutic sessions. In all the user testing we needed precise indications about the application and a general exploration of the setting.

In the following paragraphs we will report mainly from the analysis of these two sessions (Session A and Session B). Among the numerous user sessions in the water, these two sessions have been held with the current implementation of embedded non-architectural tiles. They are particularly relevant because Active Surfaces have been introduced in a structured practice and have been explored toward therapeutic objectives. These sessions represent a realistic adoption of Active Surfaces in a real setting.

5.4.2.2 Architectural simulator used by the therapists

Throughout the user testing experiments, the Architectural Simulated prototype (Par. 3.4.2.1) was adopted in two sessions at the D.Chiossone rehabilitation centre in Genova. In these trials we involved two therapists and allowed them to try out the simulator in sessions that took 30 minutes each.

The tests were organized in three phases.

(1) At first the therapists were introduced to the Tiles Simulator and were shown how it works. The researcher described the application environment and explained how the prototype works by showing the Fish Puzzle game. He demonstrated how to create an assembly through the videogame-like interface of the simulator and how it is realized by the software architecture below.

(2) Then, in the second phase, the therapists were asked to create their own assemblies. The researcher changed the game surfaces, i.e. passing from the Fish Puzzle to the Domino surfaces, and asked the therapist to repeat the task with the new game configuration.

(3) In the third phase, the therapist could freely explore the environment and create games. Each phase took about 10 minutes.

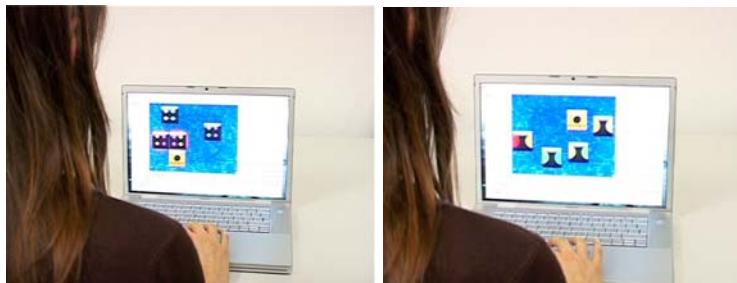


Figure 74. Therapist using the Simulated prototype.

These trials aimed at testing the fully functioning architectural prototype in its simulated application, without the constraints caused by the physical environment and the demands from the real users. The users had the opportunity to see how the PalCom assemblies are realized and how the future Active Surfaces system could function as soon as the hardware development advanced to improve the performance of the tiles. The testing with the simulator served also to envision future scenarios by relying on the potentials of the whole PalCom architecture.

5.4.3 Results

5.4.3.1 User testing in the swimming pool

The results of the user testing carried out in the swimming pool with the Lightweight embedded prototype have been grouped into main issues such as Construction Play, Re-configuration and physical programming-by-example, Debugging and General exploration of the system. These qualitative data have been determined to be in strict correlation with properties of the tiles emerging through the use. We assume that data related to such issues might contribute to the exploration of the Qualities for palpability.

Construction play

- Throughout all the testing, especially in test (1) the users looked for physical ways to connect the tiles, e.g. through physical snaps. In test (3) it was noted that, due to the waves created on the water's surface, the tiles easily detached and floated far away. Users that had impairments in handling and manipulation managed the tiles with difficulty. In session A the child put two tiles together by positioning them in the correct way. They provided light feedback and the child expected that they would also have exhibited a physical force to accompany the positive action, such as an attraction or rejection due to magnetic fields. When using a system that allows physical construction, the users expected to receive a feedback that impacted on their sensorial perception. We observed that they paid more attention to physical strengths rather than to visual feedback.



Figure 75. The therapist introduces the Domino game and starts to play together with the child.

- The quality of the light feedback was improved several times in order to reach a level that was optimal for as many users as possible, i.e. we passed from one to 5 powerLEDs to 10 LEDs strips. In the early test (2) with one LED the feedback was clearly too hard to be perceived and thus guide the construction.

Re-configuration and physical programming-by-example

- In its normal functioning, the prototype can take 5 sec to configure 3 tiles in a sequence. In 10 sec the therapist managed the physical programming and the game surfaces. When the tiles are correctly aligned they find each other within 2 sec and, when they are no longer aligned, disconnect and turn LEDs off in 2-3 sec. In Session A, due to the inconsistencies of the prototype, the physical programming-by-example once took more than a minute because of the different attempts made with the Assembler Tile via IR communication. The child started to call the therapist that was involved in configuring the tiles to ask for her assistance. In the same session, due to IR message circulation among the tiles, the programming was delayed once again because too many messages were being sent over time. It was taking too much time and, after 24 sec, the therapist had to abandon the system and went back to taking care of the child. The optimal response time of the system should be no more than 20 sec in this situation. In session B, due to the inconsistency of the prototype, programming the tiles required a number of attempts that took more than 2 min to configure the tiles. Later in the session two attempts with the Assembler Tile were necessary to configure the tiles. It took 1 minute and the therapist asked the researcher for help.
- In designing the games, the sequence and the game surface have to be configured accordingly. When ordering the pink circles (test (4)) from the smaller to the bigger, the game is correctly performed when the tiles are in the right position in any possible side connection. In (4) the system was not properly configured for this game and this caused a discrepancy throughout the interaction. The side connections were fixed while the tiles could be oriented in all four directions.
- In session B two different conditions were experimented with: at first the therapist preferred to introduce the tiles turned off and then, only after 5 min, to turn them on and configure them. Through the initial exploratory phase the therapist progressively augmented the stimuli and the complexity of the game. He gradually added the game surfaces, then added functionality by turning on the tiles, and then added more tiles: after beginning with 2 tiles he ended with four. The child actively participated in the game by asking the therapist for tiles and game surfaces.



Figure 76. The therapist prepare for the game

Debugging

- In session B the IR messages circulating along the different modules block the communication among the tiles. When the messages were not scheduled in an effective way the tiles needed to re-boot. In the current implementation it is possible to do this only by opening the upper tiles and directly restarting the microprocessor by operating on the hardware. This made the tiles difficult to manage without any technical skill.

- In session A the therapist adopted the Assembler Tile to read the communication messages sent by the tiles. She had troubles while re-configuring the tiles and, before asking the researcher, she tried to understand what was happening on her own. She inspected the system to try and understand if there was any malfunctioning. She easily read the IR messages through the LCD display in the Assembler Tile and re-configured the game sequence by giving the command again.

Exploration of the embedded system

- We collected evidence (session B) of consistent and coherent feedback, i.e. at least 1-2 sec while connecting 3 tiles and 1-2 sec for turning the LED feedback off when disconnecting the tiles. In session A there was one missed light feedback during an intense phase of exploration / discovery by the child. She thus lost interest and abandoned the activity. The LEDs turned on with a 45 sec delay. The therapist tried to present this event as being related to the previous connection action made by the child. In session A, contingent feedback (1-2 sec) gave meaning to the interaction games and captured the child's attention. The tiles showed good performance in distant communication. Even at 50 cm distance the tiles received incoming messages and provided feedback. It also allowed dynamic games involving more users at the same time.



Figure 77. Children exploring the tiles. On the Left, Session A. On the Right, Session B.

- In Session A, requirements with relation to physicality emerged. The child tried the tiles out by throwing, pushing, rotating, beating and composing them. In session B there was evidence that the child liked to put the tiles close to each other. As they are currently realized, the tiles have a rigid structure, i.e. a fixed top and bottom. This provides restrictions for possible physical combinations. In fact, apart from the side connections the current prototype doesn't allow any other interactions. It was noted that the rigid materials, which characterize the tiles, are unpleasant to come into contact with. They could even hurt the body of the child through the play activity. For the child the tiles are difficult to handle and difficult to move. The therapist has to physically handle the tiles both to program and to use them through the activity. In session B this caused difficulties because the therapist also had to hold the child. It proved to be very hard for the therapist to manage all of these things.
- In test (4) we tried the LightUp game with 4-5 year old children. They were very engaged in playing with the tiles, both moving and competing in social games. The tiles were used in competitive and collaborative games. In test (4) children also put the tiles under water and played basic water games.
- In session A, we observed the balance necessary between the rules of the game and the pleasure of spontaneous play actions. Free initiative has to be left to the children and when they were pushed too far the game was left unattended and the child preferred to do the opposite.
- In the therapeutic setting it is also important to consider the relational and emotional aspects of the interaction. In session A, the child showed her need to express her emotions and thus provoke a

reaction from the therapist. She also liked to modify the environment with her actions and create effects in the water.

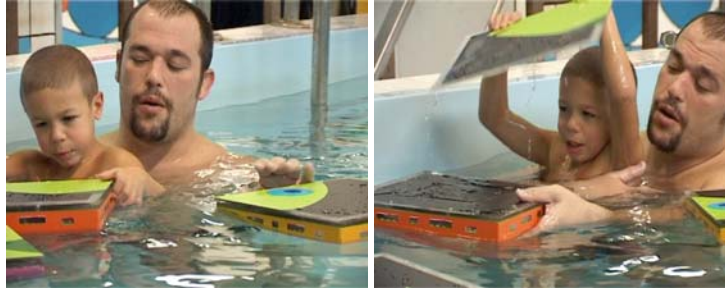


Figure 77. The child explore the tiles in Session B.

- With respect to the toys commonly used (such as the mill and the sprinkler) the tiles seemed to provoke the child's interest for shorter periods of time. They offered poor mediation between the therapist and the child, who can accomplish the tasks together, either competitively or collaboratively, but can't use one or more tiles to affect another person; rather, they used common toys to create an effect, i.e. the child used the sprinkler to make the therapist have a shower. The child is much more engaged with water responsive toys than with the tiles. In the current design implementation the tiles don't take advantage of the potentials offered by being in the water, such as being submersed in the water or absorbing and releasing the water.

5.4.3.2 User testing with the simulator results

Testing with the simulated tiles provided data related to how the assembly might be used and implemented in an embedded application. The qualitative results of the two trials (session A and session B) are described below under the following main issues: Playing with assemblies, Creation and dynamic change of assemblies, Game management and General exploration of the environment.

Playing with assemblies

- Throughout the experiments and the participatory sessions end-users have tried to compose and recompose devices and services by using the different embedded and simulated Active Surfaces prototypes. In such trials composition has been carried out without any programming skills whatsoever.
- In both the sessions the users easily understood the core functioning of the prototype during the first phase. Surprisingly, even without any specific technical skill, they easily captured the concept of the assembly and how it has been supported by the architecture. The use of the simulated Assembler Tile made this task easier to accomplish. We noticed that they perceived the assemblies and wanted to personalize the simulated environment both by changing the game surfaces and the number of tiles.

They appreciated that by both changing the game surfaces and adapting the sequence of the tiles they could easily change the game.

- Being a 2D simulation of the tiles, in session A the therapist admitted her frustration for the restricted interaction opportunities the prototype offers.

Creation and dynamic change of assemblies

- In both the sessions, during the second phase the users repeated the assembly task without any difficulties. They reproduced the actions seen through the first phase of the trials and rearranged new games by creating tile assemblies with the Assembler Tiles.

Game management

- In session B, through the free exploration phase the therapist learned to read the assemblies through the command line in the Eclipse environment. He has already experienced software application and services.

Exploration of the simulated system

- In session A, the therapist involved in the trials, who also had experience as a psychomotor therapist, suggested the use of the simulated prototype with disabled children (as end-users) within a therapeutic session outside the water, i.e. at the rehabilitation centre. It was thought to be a valuable software resource for the play activity held at the desktop computer.

5.5 Summative discussion

In this summative discussion we want to review and discuss the results related to the experimental objectives (see Par. 5.2). The experiments with Active Surfaces aimed at evaluating the user experience and the efficiency and stability of the application prototype's current implementation within the framework of Palpable Computing.

This phase constitutes the last iterative cycle of the process that is carried out in this research (see Par. 1.3.2 for details). We have described the interconnected and parallel processes consisting of the development of architectural and non-architectural prototypes (see Chapter 4) and the exploration of a range of practices (see Chapter 2). Throughout this phase the Architectural Qualities have been used as a *backbone* for both the experimental design and the evaluation framework. The results presented in Par. 5.3.3, regarding the performance of the PalCom architecture in small devices, and in Par. 5.4.3, regarding the field exploration of prototypes, allow us to discuss:

- the future development of the Open Architecture (Par. 5.5.1)
- the conceptual exploration and refinement of the architectural Qualities (Par. 5.5.2)

5.5.1 Future development of the Open Architecture

The Active Surfaces application prototypes were used to experience and evaluate the PalCom open architecture *in use*. We can thus enter into details regarding tasks and, generic and behaviour.

Both the results from the performance testing and from the user testing demonstrate that communication and discovery require optimization. As was also described in the Qualities Scenarios, Communication and Discovery Tasks revealed that performance on the embedded architecture is close to what is required from the therapeutic practices and from the requirements we got through the field experimentations with the embedded lightweight prototype. This is especially true if we consider two tiles communication (Put Together tasks). In fact, current PalVM-release performance may impact on the core practices, having an average of 3.5 seconds for a response between two tiles compared with the optimal response time of 2 sec. Such communication time responses are also crucial for providing consistent and coherent feedback, which is indeed dependent on the messages exchanged between the tiles. The results obtained by the use of 3 and 4 tiles are still a bit far from the optimum, i.e. an average time response around 7 sec. The performance of the prototype in the Put Apart tasks is still far from the optimum, an average of 10 sec against the 2-3 sec required. Improvements in speed should be required in future releases of the PalVM for UNC20-like microprocessors.

Whether it is realized through 'physical programming by example' solutions or 'remote control' solutions, re-configuration should also be optimized to realize the purposes of the palpable use of computing. User and

architectural tests regarding programming and re-configuration reveal that improvements have been done between the third year PalVM debug version (20 sec to start a service and delayed responses when more than two services run simultaneously) in comparison to the final PalVM release (10 sec to start a service).

Moreover results coming from the performance tasks confirmed that the current hardware implementation restricts the overall performance of the tiles, in particular with relation to communication.

We may conclude that the part of the open architecture experimented with the embedded architectural prototype, i.e. the PalVM and Communication&Threads, has a good basis for future development. It will still be necessary to solve performance problems related to time responses (see Par. 5.3.3), which mainly result from memory leaks in the PalVM performance and bandwidth limitation in the IR communication module.

With PalVM optimization for UNC20-like microprocessors and the VM running directly on hardware support, without the uCLinux OS, it would be possible to save space and memory. Such improvements would allow us to adopt the middleware managers (or part of them) and the PalCom assemblies.

It could also be possible to augment the computational power of the executing platform to achieve a better performance. Although it would provide resources for the application prototype, it would also be outside the scopes of this research. In fact, the main goal of the application prototype is to optimize the PalCom Ubiquitous Computing open architecture to fit into resource constrained embedded devices and to reach acceptable performance standards in relation to the application prototype. In fact the main challenge for the Active Surfaces application is scalability. The system is currently implemented in four units but in future implementations we expect to produce a higher number of tiles, thus further challenging the architecture. The comparison among an increasing number of tile configurations (2 Tiles, 3 Tiles and 4 Tiles) suggests that the performance of PalVM and the PalCom communication components should be improved to meet the requirements of a scalable system and guarantee acceptable time responses as the number of the modules increase.

Scalability also demands a dedicated communication model that may fit with distributed assemblies or groups of assemblies as embodied by the Active Surfaces conception. The periodicity of the devices also has to be taken into account: every device has a periodicity related to discovery and communication. In horizontally scalable systems the individual internal timing of each tile poses challenges of synchronization and periodicity that the PalCom discovery protocols have to address.

In what follows we outline the possible and auspicious uses of the open architecture in Active Surfaces:

- PalCom assemblies.
- Assemblies of assemblies.
- Exploiting the Resource Manager to the utmost (2nd order resources).
- Contingency Manager, in order to achieve System adaptation.
- Storage component, in order to realize Re-use.
- Assembly Browser

As examples we address in more details the potentials that the Storage component and the Assembly Browser may have if used in this application prototype. Although they are not yet been implemented, they maybe very relevant to the Active Surfaces application because of the valuable user and system opportunities they might provide. These potentials will also described and explored by means of Prototype scenarios, which has been evaluated with the users, and may contribute to the future development of the architecture and application.

5.5.1.1 Browsing, Storage and Re-use

A possible exploitation of the PalCom Browser in the Active Surfaces scenario could be very beneficial for all the practices of planning and evaluation of the therapeutic activity. The PalCom Browser is a GUI that has been developed to allow the direct inspection and management of 2nd order resources. It allows service composition and creation, construction and deconstruction of assemblies. A number of versions of the Browser exist and a dedicated release could also be designed and adapted to the tiles. The Active Surfaces Browser either could be used in conjunction with the Assembler Tile or with alternative programming tools to allow the users to manage the 2nd Order resources. It can be used in conjunction with the Storage component to allow a management of the system that is simple and meaningful for the therapists.

The Storage Component is part of the Utilities that have been developed together with the architecture. It provides a simple means for making storage possible: through a common interface, various media could seamlessly and continually be stored on a file system or flash storage. The PalCom Storage Component handles various aspects of file storage: persistent data, simple database transactions, searching and indexing of data, scalable data structures and safe data handling with encryption (PalCom Deliverable 54). The Storage component also has the potential to handle multiple and simultaneous storage repositories. This facilitates distributed PalCom applications ability to construct and deconstruct data from different sources and devices.

The Storage would allow the opportunity to store and re-use previous resource configurations that is a valuable aid when some solutions prove to be successful in particular cases and they are ideal candidates to be re-used in similar situations.

Re-using successful solutions and keeping track of the best practices is realized by the collection of the past activities. People need both to adapt and stabilize the practices they make sense of. The memories created are enriched by personal values (e.g. adding notes, clustering, tagging) to be better recognized and used. Unnecessary repetition of procedures can cause great discomfort and damage.

In the following section we describe Re-use by mean of scenarios, firstly describing the current practices around this issue and then describing how the activity may change with the introduction of palpable technologies. The Activity Scenario 9 is the continuation of Activity Scenario 1 (Par. 2.4) regarding the end of the activity in the water and the evaluation. This scenario has been developed as outcome of the activity analysis described in Chapter 2.

Activity Scenario 9

Repository of traces

The session with Andrea went really well and the water therapist Alessio followed the activities he has envisioned during the planning stage. He is currently evaluating the activities that have been carried out. In a certain way Alessio has tried to keep in mind the relevant points throughout the activity itself. In fact, he has repeated the successful games throughout the reinforcement phase. In this phase Alessio has tried to stabilize the acquired results and to provide the child with a positive conclusive feeling before leaving the water.

Now that the session is finished, Alessio is at the rehabilitation centre and tries to take notes of the relevant events in which Andrea has taken part. The therapist makes particular reference to the therapeutic objectives and tools adopted throughout the activity. Andrea gave meaningful responses regarding attention and the reciprocal exchanges mediated by the use of the sponge. He succeeded in maintaining, shifting and sharing his attention for the period of time required by the game. Alessio has noticed that dividing the sponge into two parts better supported the imitation games than the use of a single piece.

Alessio writes down such remarks and also describes the results regarding the physical activity and the appropriateness of the tools used (e.g. the set of noodles for the symbolic game).

Alessio keeps all the notes on the sessions together in the same block and uses this repository as a history of the treatment. The descriptions he wrote are traces of both the best practices and the advancements of the child. Alessio will use his notes on the current session as a basis for planning the activities for the next one. Such traces are also used for periodic and summative evaluations of the therapy.

Keywords: Traces; History of interactions; Re-use successful solutions; Collecting best practices.

The therapist practice “Reuse and keep trace of the best practices” is the possibility for defining the best practice and for reusing a configuration of objects and settings, creating a toolbox containing the know-how concerning creative usage of the existing therapeutic tools.

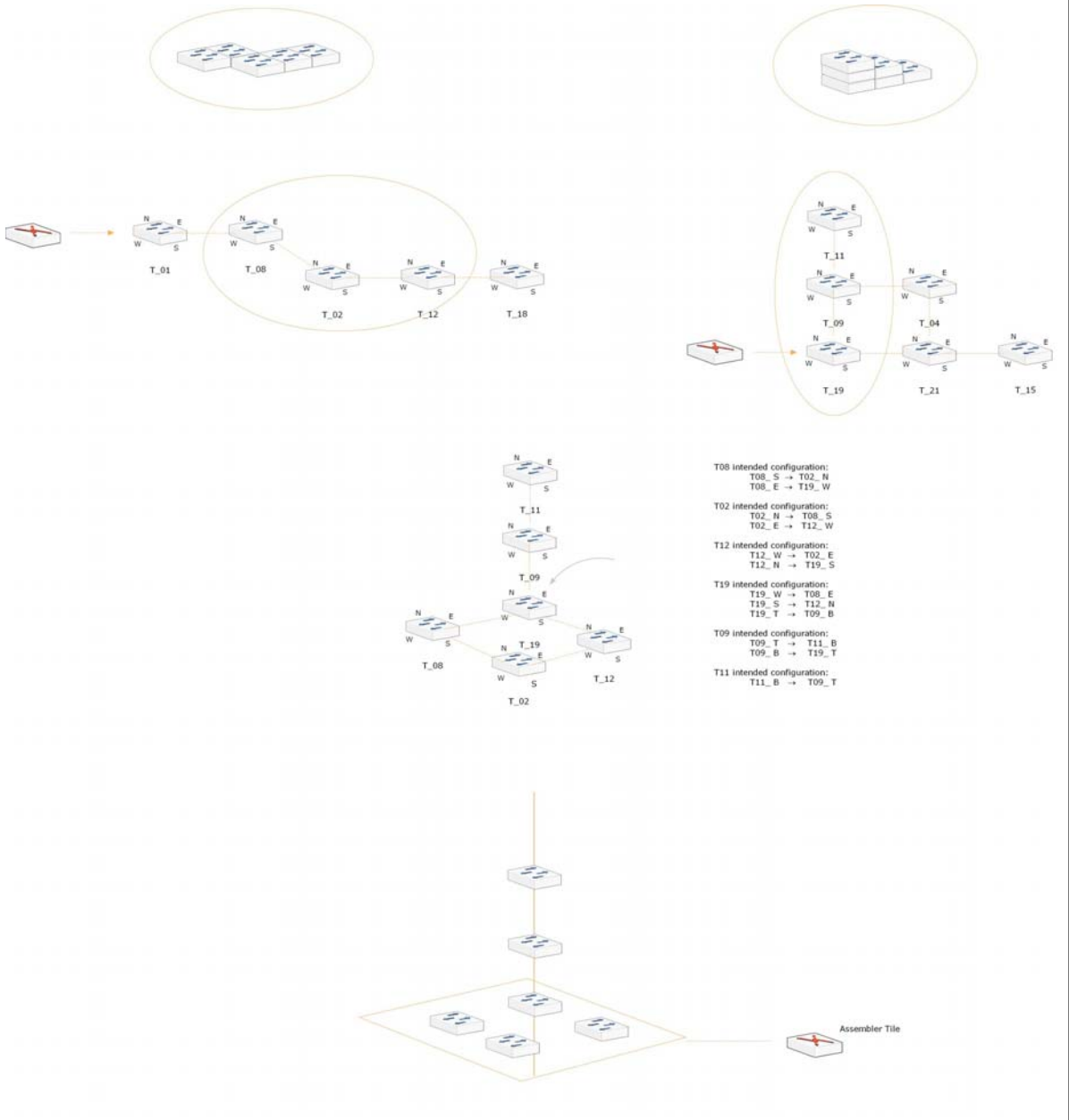
By using the Storage component, Active Surfaces may supports the re-use of a previously instantiated configuration of resources by recording the data about the tiles’ Assembly that have to be saved and stored. The storage may be supported by the use of the Assembler Tile. The Assembler is currently though of as a tile that differs from the normal tiles since it is the tool used to program the tiles, create the assemblies and debug the connections. It may also be designed to support storage and re-use.

The following Envisioning Scenario describe how the current practices may be modified by the introduction of the envisioned architectural solution.

Envisioning Scenario 3

Saving and re-using assemblies

Now that the session is finished, Alessio, a water therapist, is evaluating the activities that have been carried out. He wants to save the specific configuration of Active Surfaces that he has used. He uses the Assembler Tile to save the GameAssembly, he downloads the file by connecting with one tile of the assembly. He has now saved the physical and the logical configuration of the game that has been used, together with a description of the results on the AT (see the figure below, on the left). In the following session Alessio will use part of this assembly as a base to be combined with another GameAssembly that was previously saved. He is planning to reuse two parts of the assemblies present in the Assembler Tile repository in order to create a new GameAssembly.



5.5.2 Architectural Qualities explored and revised

The possible architectural development cannot then be fully foreseen in practice without referring to the Architectural Qualities that must be implemented. As we briefly discussed above there might be several ways to embody and realize the Qualities for palpability. In fact, each Quality is an instance of the dialogue between the Application domain and the Architecture domain. These two fields of research correspond to user decisions in the case of the former and to middleware managers' decisions in the latter.

The Qualities constituted a way of translating the concept of *palpability* into more operative attributes that helped in the conceptual and empirical exploration of this notion. We were interested in understanding how *palpability* might contribute to current Ubiquitous Computing technologies and, in order to do it, we have detailed aspects of palpable computing and bridged the Application level and the Architecture level.

As described at the beginning of this chapter (Par. 5.1) the experiments with the Active Surfaces prototypes were organized around the Qualities. In particular, we tried to further translate the Qualities into low-level issues that could be directly afforded throughout the experiments.

The experiments resulted in quantitative and qualitative data that directly informed the following issues:

Assemblability:

- Construction Play
- Playing with assemblies

Adaptability (as User adaptation)

- Re-configuration
- Re-configuration and physical programming-by-example
- Creation and dynamic change of assemblies

Adaptability (as System Adaptation)

- Negotiation and system intelligent adaptation

Resource awareness

- Communication and discovery
- Debugging (1st Order Resource awareness)
- Game management (2nd Order Resource awareness)

Experimentability

- Performance
- Exploration of the embedded system
- Exploration of the simulated system

These punctual issues allow us to revise the Qualities with relation to the Active Surfaces scenario. Since the prototypes embody each individual Quality in a specific manner (see Par. 4.3), when we experimented with these applications we realized the Qualities within the specific context of the Active Surfaces case study that is different from all the other application prototypes developed in PalCom.

The results of the experiments and the scenarios development allow us to revise and elaborate on the initial formulation of the Qualities (Chapter 1).

The Architectural Qualities have been revised as follows.

Assemblability

The initial formulation of Assemblability is inspired by the modifiability quality attribute of Bass (2003) and the challenge that palpable systems should complement the construction of systems with deconstruction.

Assemblability is related to heterogeneous as well as homogeneous devices that are assembled in different formations. The latter refers to the modular Ubiquitous Computing system in which each unit has the same resources and functions. Each tile constitutes an assembly in itself and can also

be put into assembly with other equal devices. . In the second case we have assemblies on two levels: on the lower level each device constitutes an assembly, which is part of another assembly that is made of assembled constructs at the higher level (Assembly of Assemblies). We can think of the low level assemblies as being managed by the primary users, e.g. the therapists in our scenarios, and the high level assemblies as being part of the activity with the secondary users, e.g. the children with special needs.

The assemblies formed at the higher level can be the focus of the activity in and of themselves. They are not used to reach an ultimate goal, but the scope of the activity may be to play with the existing assembled formations by allowing physical and logical construction, deconstruction and reconstruction to be carried out manually or automatically. Users playing with the assemblies of devices need means for construction that are coherent with the activity undertaken. Without any knowledge about low level assemblies (e.g. each tile constructed as an assembly of services) users continue to easily perceive and understand high-level assemblies (e.g. the system made of different tiles). Since Ubiquitous Computing technologies are not immaterial, but rather are embedded resources with a physical nature, users need to physically control and organize the assembled constructs.

Adaptability

A system exhibits adaptability if it is possible to change the behaviour of the system by replacing or altering the behaviour of some of its elements without affecting the rest of the system.

It can be carried out as a dynamic resource reconfiguration that the system autonomously carries out (System Adaptation) or can be effected through human interaction and user decision (User Adaptation). We experimented with User adaptation by observing how the assembled resources can be modified by manual changes and how the users manage and flexibly adapt the system configuration to their needs. Flexible adaptation results in the tailoring of computational resources, either physical and software Devices or Services. Adaptation should be a gradual and progressive process as is required by the use practice.

The dynamic change of assemblies needs to be consistent from a user perspective. It should regard easily perceivable and intuitive manners in order for the technologies to also be palpable from the perspective of user interaction (e.g. the physical programming-by-example we have experimented with by developing the Assembler Tile prototype). Palpable technology has to encourage users to make changes and prevent disuse and abandon.

The changes being made at the user level should reflect those that happen at the functional level. Users need to perceive that their decisions impact the system functions as they are performed at the user level, i.e. the system should reflect the user's choices.

Resource Awareness

In its initial formulation, Resource Awareness describes the fact that resources are aware of one another's presence, availability and behaviour. Being aware of the presence of another resource implies the ability of one resource to discover other resources, whereas being aware of the behaviour of another resource implies the ability of one resource to determine the operational capabilities of other resources (PalCom Deliverable 54).

Present resources have to be discoverable in a way that is easily perceivable by the users. In this way palpable technology should help users to *touch* the resources they can use, i.e. understanding which resources are available and discriminating between hardware and software resources. A proper understanding of the resources in use affects the way in which users make decisions about the system and thus about the activity, e.g. the therapist realizes that the IR communication modules have a degraded function and decides for a different game activity that doesn't exploit communication among tiles.

When possible, debugging and recovery should also be supported for users without any technical skill, e.g. the therapist adopts the Assembler Tile to *read* the behaviour of the system and can recover part of the eventual malfunctions. Resource monitoring and management have to be supported over time. Users need to continuously read the behaviour of the system.

Experimentability

Experimentability describes the capability of facilitating and encouraging exploratory experimentation by users participating in systems that conform to palpable computing principles. Because they are error-tolerant systems, both simulated and embedded assemblies of devices can be used, and are customized and altered within certain levels of freedom and performance.

Users can explore palpable technologies and take free initiative. In this way, new possible use destinations can be envisioned with relation to the users' backgrounds, attitudes and objectives. Throughout exploration users might discover usages that are meaningful for them and unexpected for the designers.

As it is an Ubiquitous Computing technology, the exploration of the palpable system is affected by the physicality of the devices and the fact that they are a part of the environment. The users may perform physical actions that might or might not affect the behaviour of the system, e.g. throwing, pushing, rotating, beating and composing the tiles. Experimentability also concerns the role that palpable technology plays within the environment. In fact Palpable computing may depend on the environment, may make use of environmental resources or can be affected by changes occurring within the environment. Through field exploration the relational and emotional aspects of the interaction also play a role, e.g. the light feedback used can evoke something that is not pleasant or desirable for users.

Scalability hasn't been specifically addressed by individual experiments, but rather it is a quality that all of the prototypes realize in different ways (see Chapter 3) and that is investigated in a transversal way. In fact Active Surfaces embodies some of the aspects of vertical and horizontal scalability that can be described through architectural qualities like Assemblability and Experimentability. They together support us in discussing the horizontal and vertical Scalability of the PalCom open architecture.

In particular, by testing the Active Surfaces prototypes, we have addressed the Experimentability and exploration of resource-constrained devices. This resulted in the performance testing carried out with the current hardware implementation. Active Surfaces allowed us to evaluate the vertical scalability of the open architecture and the concrete opportunities to realize scalable Ubiquitous Computing applications (Par. 5.3.4).

The exploration of Assemblability, instead, supports us in discussing horizontal scalability and modularity. By exploring the Assemblies of assembly we might have an architectural and logical model that provides support for horizontal scalability. Such an architectural concept allows us to give a functional structure that enables modular, distributed and ubiquitous computing.

We conclude that the Qualities are valuable concepts used to describe the open architecture and also valuable tools for the conceptual exploration of palpability. They have been used to summarize the iterative process and to inspire and guide the experiments.

The Qualities can then be implemented on several architectural levels: the IR communication code is split into two parts, whereas the actual IR code that talks with the hardware resides in codebase, and the handling of the receipt and sending of messages over the PalCom protocols resides on the IRMediaManager part of the Communication component. The multi level instantiation of the architectural Qualities demonstrates that they emerge as interwoven properties of the open architecture that can be distinguished on conceptual and

experimental levels. The research presented in this thesis can be considered an example of this process.

Finally, among the findings of the research we also collect indications to re-design and further develop the Active Surfaces application, mostly with the user testing. The tiles have to be improved from a design perspective as well. We gained indications about the tiles' physicality (materials, weight, dimensions) and interaction modalities. We observed that the current design doesn't allow for the full exploration of the potential of the concept with relation to the aquatic environment. We have also envisioned possible future developments of the concept through which supporting further exploration of the water environment and of the therapeutic activity. In the figure below we tried to visualize the envisioned development of the concept.

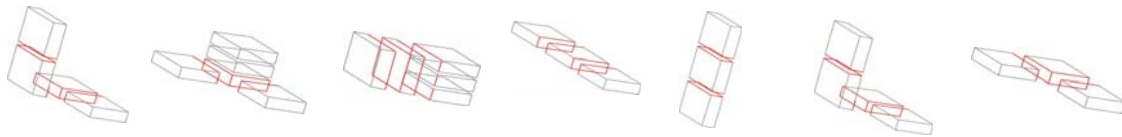


Figure 75. The Active Surfaces concept.

As shown in the figure above the future development of the concept may be based on the construction of physical components in various and heterogeneous formations that explore all the possible connections along the vertical and the horizontal dimensions.

The tiles physical case should be re-designed to support discovery games in the water or to be placed on the bottom of the pool for the creation of water carpet-like configuration. We also critically noted the meaning that light feedback has with relation to physical attraction-rejection among the tiles. We observed that, since the tiles are construction modules, the forces created by magnetic fields supported the user interaction better than LED light feedback. Users seemed to be more inclined to follow the guidance provided by the magnetic forces than the light.

Together with architectural development, the further re-design of Active Surfaces is needed to support users in experiencing this UbiComp technology in a palpable way.

Chapter 6

Conclusive Discussions

The conclusions of the thesis aim to discuss the research objectives that motivated this work and the accomplishments we have achieved throughout the whole process. This chapter is in fact organized in three paragraphs, each of which deals with one of the initial research objectives, Par. 1.1. At a conceptual level, this thesis aimed to investigate the user experience within the Ubiquitous Computing paradigm, how people use UbiComp resources and how their adoption impacts everyday practices (Par. 6.1). This was specified as a more operational objective: the exploration and experimentation with the PalCom open architecture in resource constrained embedded devices (Par. 6.2). The third methodological objective regards the way in which the use of different application and architectural prototypes inform the development of Ubiquitous Computing architecture (Par. 6.3). At the end of the chapter open questions and research perspectives will be presented (Par. 6.4).

6.1 User experience and UbiComp

The visions of *invisible* (Norman 1998), *disappearing* (Wejchert 2000) and *calm computing* (Weiser 1998) have in common a strong focus on how users experience distributed, networked and interactive technologies, and how the environments might be augmented by ubiquitous technologies. Within the present research we found that the support of architectural concepts and structures is needed to achieve the visions presented in Chapter 1 and to enable the resulting user experiences.

This field still lacks definitive methods of addressing issues such as the role of the users and the qualities of the interaction. People using Ubiquitous Computing technology are assumed to be on one hand experienced users with technical knowledge, on the other hand they are defined in abstract, generic, and immaterial way. In fact, in some cases they are required to hold technical skills, while in others they are completely unaware of the processes that they are part of. Thus the interaction design results in underestimation of both in its potential and the challenges it might provide (Suchman 2007).

Traditional engineering approaches, which are mainly oriented to the definition of functional requirements, seem not to be efficient enough to perceive the complexity of the interactions within UbiComp (Abowd, et al. 2002). It seems that we go back to what happened with early interactive systems: there seems to be a lack of consideration of the aspects of human behaviour, of the interaction with other people and with the environment, of the historical, personal and cultural background of the users (Norman 1980).

In Active Surfaces we have an example of distributed, embedded and modular technology that is useful for investigating this paradigm from user perspective. With this thesis the concept of the *palpable* use of technology is contributed, as described in Par. 4.5 through the Qualities for palpability.

The specific focus on *people using technology* is at the centre of the *dialogue* between use and architecture and this research proves to be one of the few empirical studies on performing and evaluating UbiComp technology with relation to the end-users (Rodden et al. 2004; Bellotti, Edwards 2001; Halloran et al. 2003; Abowd, Mynatt 2000; Abowd et al. 2002; Newman et al. 2002; Markopoulos et al. 2004). Much of the literature of this still rather young domain consists of few systematic works on usage. In this research we aimed at contributing to the analysis and evaluation of Ubiquitous Computing *in use*, in particular how it might impact human activities in the authentic contexts of use (Abowd et al. 2002).

The research investigated the key *use practices* of therapists working with people with special needs, who are our primary and secondary users, respectively. In this investigation we followed an ethno-methodological and empirical approach that considers every course of action as essentially dependant upon its material and social circumstances (Suchman 1997). We observed and analysed the relations and the interactions among our users, and between them and the artefacts they adopt (Chapter 2).

The environment within which the interaction takes place also necessarily affects the performance of Ubiquitous Computing. In our case study, Active Surfaces allowed us to observe the mutual exchange users have with the water environment. Water simultaneously forces people to adapt and enhances the available potential for action (Hornecker 2005; 2006; 2007). It is also a special environment that react to user actions in a contingent way.

People with special needs have different potentials for action with relation to the environment and the different kinds of available resources. Furthermore, UbiComp resources might be used to enhance the potential for action of people with special needs. These resources have a computational and a physical-material consistency: they are computational resources that enter the physical environment by being embedded in powerful distributed devices.

Many of the resources available in the current Ubiquitous Computing environment consist of complex, layered and abstract processes that are immaterial, hard to perceive and hard to manage. In Active Surfaces we tried to support our users as they attempted to make sense of ubiquitous computing. They meaningfully adopted the tools that made sense to them and by means of physical acts (cfr. Shafer 2001). User actions demonstrated to be purposeful and opportunistic (cfr. Abowd et al. 2002). User practices are opportunistic in the sense that they are directed toward objectives, oriented toward effective performance and efficient use of resources. *Opportunistic action* is part of the natural and economic strategy people enact when taking advantage of the opportunities the environment provides in order to pursue personal goals and save resources. This was especially evident for people with limited abilities in the variable water environment. The use of Ubiquitous Computing was thus ruled out by opportunistic strategies, which can be explained as an optimization of the resources available in the environment through which the users aim at avoiding frustration and danger and seek pleasure and safety.

Interpersonal relations also played a central role within this case study. The therapists had to make their actions on technology meaningfully accountable for the children. They had to make sense, inspire and motivate play and support children throughout the activity. This was achieved by negotiation and mutual tuning, and by means of adjustments and adaptations through which the therapy is structured.

This study helped us to figure out the *interplay* that is needed between use and architecture. We observed that the introduction of UbiComp technology affected and changed users' activities and that, at the same time; they became responsible for maintaining, controlling and changing it. The system *architecture / use* relationship is dialectical since on one hand, technology enhance certain practices by enabling novel use opportunities, on the other hand user-specific dynamics provoke, inspire and inform the emergence of unpredicted architectural solutions.

6.2 Active Surfaces as *toy problem*

Within the PalCom project the Active Surfaces application prototype can be considered a *toy problem*, through which the development and experimentation with the Ubiquitous Computing open architecture progressed throughout the project. In fact, the Active Surfaces scenario didn't pose immediate research interest, but rather it was appreciated as an expository problem to illustrate traits that were shared by other scenarios as well as more complicated instances of the problem, i.e. the scalability and use of embedded resource constrained devices. Active Surfaces represents an oversimplified case of the scalability problem that challenges the architectures for UbiComp.

The PalCom project was oriented to the development of a middleware architecture and it was organized in sites deputed to software development and others deputed to application scenarios and user studies. Active

Surfaces was selected among the health care application scenarios. As described in Chapter 3 the tiles provided to be a test bed for the experimentation with the open architecture in resource constrained embedded devices. In fact Active Surfaces has been a valuable scenario for testing part of the architecture in UNC20-like microprocessors.

Active Surfaces results then in a *toy problem* that allows investigating scalability, portability, demonstrability and flexibility. In this thesis the concept of scalability in Ubiquitous Computing is empirically explored. Here scalability is introduced as a property of the Active Surfaces concept and it is then discussed considering the structural value it may provide and how it can be achieved at the architectural level. Because it is a scalable modular system, Active Surfaces was useful for demonstrating the potential of the architecture in a way that was immediately comprehensible. The tiles prototypes were indeed very powerful in demonstrations and exhibition. As example, because it has a videogame-like appearance, the Simulated Architectural prototype realizes Active Surfaces as toy problem in a very special way.

Active Surfaces allows for the direct *experimentability* of the architecture, it is made immediately available for users. Together with the other architectural Qualities embodied by Active Surfaces, experimentability provides users with a system that they can play with in very literal way. For its simplicity, it offers the opportunity to be easily explored and evaluated. It can be horizontally and vertically scalable and this scalability implies an ability to efficiently manage the distribution, availability and configuration of resources according to prevailing activities and objectives. By allowing this, Active Surfaces can adapt in form and function according to changes in operational context and users-specific features. This ensures that it can be dynamically constructed and deconstructed, and the mechanisms that control its *flexibility* are either user-triggered or triggered by an automated mechanism.

A further effort has to be made to ensure that the *user* easily comprehends and is in control of the palpable tiles. This includes exposing operations with a visible and adjustable level of detail, dependent on actor actions and external events. The goal is to reach a point where the various users are an integrated aspect of the palpable environment rather than passive actors. In this perspective, since it is useful to illustrate the dialogue between the use and the architecture, Active Surfaces could allow improvements for the architecture to be really *palpable* for users.

6.3 Methods and prototypes

In this thesis steps are made towards understanding the nature of research in Ubiquitous Computing. Compared to other well-established domains, Ubiquitous Computing is not yet clearly disciplined. A major part of the research in this thesis aims at contributing to the design, development and evaluation of tools that support the exploration of UbiComp technology made palpable. Methods like activity analysis, prototyping, lab testing and user testing have been developed to identify architectural solutions. We tried to inform the conceptual framework and the architectural development following a bottom-up approach to the design of Ubiquitous Computing. This methodology strongly relies on prototyping and field exploration to experiment with and evaluate palpable computing, and this was a major solution for assuring robustness and foundation of the research carried out.

We contributed both describing the set of methods and techniques used and introducing demonstrators and prototypes to probe concepts, applications and architectural components of Palpable Computing. The issues presented in this thesis do not result in a coherent new research methodology. Rather, they provide a set of tools that can help to facilitate reproducible research in UbiComp.

The methodology we have followed places everyday operational practice at the centre of the research. The design and deployment of palpable computing solutions relies upon a thorough understanding of the ways in which people accomplish work and interact with each other through tools and technologies in everyday work environments. In this case study the actors were the therapists and the patients within the practical circumstances and constraints of their everyday activity.

As stated above there is not a specific methodology for evaluating Ubiquitous Computing from user perspective. In many cases rather methodology is adapted with strong relation to the background of the researchers.

As detailed in Chapter 3, a wide variety of methods have been used throughout the iterative research and development process. These methods pertain to Human Computer Interaction, Participatory Design and Software Architecture Engineering. The developmental process followed in the project is also incremental, in the sense that it continually improved the outcomes to a greater extent. Since the processes have not been carried out in a sequential way, but rather throughout iterative cycles, the achievements have been continuously evaluated, improved and re-used to feed other processes. Architectural and non-architectural prototypes embody integration among the traditional ethnographic studies, participatory design methods and naturalistic experiments to inspire, inform and evaluate the design of software architectures (Bardram et al. 2004).

We studied how the use of architectural and application prototypes could support the research in Ubiquitous Computing. In particular, we designed and deployed application and architectural prototypes to inform the development and evaluation of software architecture for making UbiComp palpable.

This investigation required design processes and research methodologies that were appropriate for the investigation of the therapeutic setting. Dealing with diverse and special users required non-obtrusive, personalized and adaptable tools. To reach this goal, architectural components were used and implemented in a number of different architectural prototypes. The PalCom architecture offered the possibility of experimenting with use solutions from different perspectives (that of disabled individuals and the caregivers) by providing a common middleware environment for solutions suitable to special needs.

The experimentation and evaluation of prototypes have been performed to identify and compare strengths and weaknesses in architectural developments during the design process. In fact the existing systems have been experimented with within real contexts and the extensive use of architectural prototypes is one of the main characteristics of the process.

Iterative processes of mock-up and prototype experimentation also characterize the overall process. Through them the open architecture has been experimented with both in laboratory software testing and in embedded and simulated applications within the scenarios selected in the project. The field experiments helped software developers make sure that the architecture would be able to fulfil the requirements of the use practices and will reify the Qualities for making Ubiquitous Computing palpable for users.

We used horizontal (the Lightweight embedded prototype) and vertical prototypes (the Architectural embedded and simulated prototypes). We followed exploratory and experimental prototyping into a co-evolutionary process, in which the prototypes were gradually developed into the current implementation, either incrementally such that more and more details are added stepwise or by iteratively evaluating and refining what has already been covered. The different prototypes have been developed in parallel since the beginning of the process so that each path informed the others without constraining them. The convergence of the parallel processes was obtained through the use of Scenarios, with the aim of arranging data gathered

through activity analysis, systems envisioning and functionalities specification, and finally assessing the implemented solutions.

The application and architectural prototypes were the means used for designing, building, and evaluating the open architecture through the investigation of the architectural Qualities for palpability. As described above each of the prototypes we have developed and experimented with embodies some of the Architectural Qualities in several different ways. The Active Surfaces prototype described in the Prototype Scenarios (Par. 4.4) has been partially implemented in the different experimental prototypes currently in existence and is also the expected result for the next phase of the design and development process.

By practically experimenting with prototypes it was possible to evaluate the open architecture in the authentic context of use (cfr. Abowd, Mynatt 2000). They provided the opportunity to conduct a methodological inquiry about Ubiquitous Computing and end-users. A combination of different kinds of prototypes supported us in creating an experimental situation within which the evaluation of the software architecture made sense from user perspective.

6.4 Open Questions

With each prototype tested and evaluated, a number of challenges arise together with open questions that haven't been addressed and have been left unexplored in this thesis. In this last paragraph we go back to one of the challenges described at the beginning of the thesis, the balance between *system automation* and *user control*, in order to discuss open research questions and future development.

System automation / User control is not only a challenge for Palpable Computing, but also affects the entire investigation of Ubiquitous and Ambient Computing. It is a central theme in Ambient Intelligence as well. It can be in fact formulated as system automation, and independence or intelligence to be compared with and tailored on the end-users needs to control, change and adapt the tools they use.

This theme resulted from the discussion presented in this research about one specific architectural Quality: Adaptability. We have presented the initial (see Par. 1.1) and the revised (see Par. 5.5.2) formulations of Adaptability, describing Adaptation based on *user decision* as well as Adaptation based on *system decision*. Active Surfaces is an adaptable and adaptive system that can be configured through time by the users and may adapt its behaviour by itself. In either an autonomous or user controlled way the system can change with respect to the changing conditions of the activity. The occurrence of one mechanism with respect to the other has to be negotiated according to the objectives of the activity in which they are used. In some cases it is desirable that the system intelligence rules out the behaviour of the system, even if based on initial user configuration; in other cases the user has to be fully in control of the system and tailor it to the needs of the patients.

Adaptability especially resides on the Contingency Manager (Par. 4.4.2) that has not been fully investigated in this research. Due to hardware constraints it hasn't been possible to experiment with the middleware managers in low resource prototypes yet. We contributed particularly by investigating the prerequisites to support Adaptability and describing the impact it might have in possible future scenarios.

Through the performance testing we explored the support provided for a dynamic re-configuration of the architectural prototypes. We have evaluated that support is provided for the future adoption of the Contingency manager. The main difficulty is the computational power that UNC20-like microprocessors should have to support the functioning of the middleware managers. This can be considered the major requirement for the introduction of the Contingency Manager.

Then we describe the impact that system automation might have in the therapeutic practice. This is achieved through the exploration of possible future scenarios where Adaptability is based on system decision. This is described in the following envisioning scenario.

Envisioning Scenario 4

Negotiation and system adaptation (Based on System decision)

Laura is in the water with Federico, a 7-year-old boy with Autistic Spectrum Disorders. From the beginning of the session Laura tries to maintain a dialogue with the child. She starts to use Active Surfaces and pushes the tiles against the other.

Federico starts to laugh; he seems to appreciate the activity and slowly starts to imitate the therapist. By simply pushing a button on the Assembler Tile she sends a wireless signal (e.g. via Bluetooth) to make the tiles enter in autonomous mode.

The tiles learn to give different light feedback with respect to input actions provided by different users. After a few minutes, as soon as Laura and Federico move the tiles, they splash each other, and the child starts to cry and shout with discontent: he really dislikes being splattered with the water.

Laura abandons this game and waits for a minute. Then she starts the DominoGame assembly by launching it with the Assembler Tile. Laura carefully introduces the tiles and the Domino game surfaces. Federico enjoys the colours of the surfaces. After 3-4 minutes the child slowly loses interest in the game and moves away.

After 5 minutes of receiving no input the Active Surfaces proactively start a new game. The LightUp game consists of the tiles alternatively blinking one after the other. The light captures the attention of the child. He follows the light and reaches each blinking tile by moving within the environment. He is clearly engaged in this game and Laura lets him playing freely.

This scenario raises questions that still have to be answered and, in order to do that, further empirical experiments about use and UbiComp architecture are required.

The balance between *system automation* and *user control* can also be seen as the need for cooperation between system and user *intelligences*. When is it desirable to have this cooperation between system and user intelligence?

How may the role of the users evolve in this mixed control of ubiquitous system?

When is it desirable that a system proactively takes initiative in social settings?

Which kind of system autonomy is desirable when we design for people with special needs?

Adaptability means that system configuration may be altered and adapted over time. How should either system or user control be guaranteed in eternally changing settings?

When is the system's autonomous comprehension of the current state of context desirable?

Is it desirable that the system holds the full responsibility of every process and choice?

In a completely ubiquitous system, which component should be responsible for the choices? Who has the ownership of these choices?

We would need a conceptual and architectural model for distributed intelligence in order to experiment with these issues. This mechanism should be able to take the initiative and be responsible for the behaviour of the whole system.

The PalCom open architecture would provide conceptual and architectural support for the investigation of these kind of questions. The use of PalCom Assemblies would provide a structural, logical and functional

unit to experiment with distributed intelligence, i.e. user-system or system-system. Moreover, the intelligence may essentially reside in the three middleware managers, Resource, Assembly and Contingency managers. Their combined use may provide an architectural foundation for the development of intelligent ubiquitous applications.

The questions outlined above represent a key challenge for current Ubiquitous, Pervasive and Ambient Computing and thus inspire future research in a direction that is promising to the improvement of this domain, in addition to also raising ethical concerns related to persons and their individual conditions of life.

References

- Abowd, G., Mynatt, E., (2000) Charting Past, Present and Future Research in Ubiquitous Computing. ACM Transactions on Computer-Human Interaction, Special issue on HCI in the new Millenium, 7(1):29-58, March.
- Abowd, G., Mynatt, E., Rodden, T. (2002) The Human Experience. IEEE Pervasive Computing, vol. 01, no. 1, pp. 48-57, Jan-Mar, 2002
- Amundsen, S.L., Eliassen, F., (2006) Combined Resource and Context Model for QoS-aware Mobile Middleware. Proceedings 19th International Conference on Archiecture of Computing Systems (ARCS2006), Lecture Notes in Computer Science, Springer, 2006, pp. 84-98
- Andersen, P., Bardram, J.E., Christensen, H.B., Corry, A., Greenwood, D., Hansen, K.M., Schmid, R., (2005) Open Architecture for Palpable Computing Some Thoughts on Object Technology, Palpable Computing, and Architectures for Ambient Computing. Ob-ject Technology for Ambient Intelligence Workshop, Glasgow, U.K. Proceedings of ECOOP 2005.
- Aucouturier B., Darrault I., Empinet J.L. (1984) La pratica psicomotoria. Tr. it. Armando, Roma. 1986.
- Bahrck L. R., Watson J. S. (1985) Detection of intermodal proprioceptive-visual contingency as a potential basis of self-perception in infancy. Developmental Psychology, 21:963--973, 1985
- Bardram, J. E., Christensen, H. B., and Hansen, K. M. (2004) Architectural Prototyping: An Approach for Grounding Architectural Design and Learning. In Proc. 4th Working IEEE/IFIP Conference on Software Architecture, pp. 15-24, 2004.
- Bass, L. J., John, B. E. (2001) Supporting usability through software architecture. IEEE Computer, 34 (10), 113-115.
- Bass, L. J, John, B. E., Kates, J. (2001) Achieving usability through software architecture. Carnegie Mellon University, Software Engineering Institute Technical Report No. CMU/SEI-2001-TR-005.

- Bass, L., John, B. E. (2002) Supporting the CANCEL command through software architecture, CMU/SEI-2002-TN-021. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University.
- Bass, L., John, B. E. (2003) Linking usability to software architecture patterns through general scenarios. *Journal of Systems and Software*, 66 (3), 187-197.
- Bass, L., Clements, P., Kazman, R. (2003) *Software Architecture in Practice*. ISBN: 0-631-21304-X, Addison-Wesley, 2003.
- Belotti, V., Edwards, K. (2001) Intelligibility and Accountability: Human Considerations Context aware systems. *Human-Computer Interaction*, Volume 16, pp. 193–212
- Belotti, V., Back, M. J., Edwards, W. K., Grinter, R. E., Lopes, C. V., Henderson, A. (2002) Making sense of sensing systems: five questions for designers and researchers. *ACM Conference on Human Factors in Computing Systems (CHI 2002)*; 2002 April 20-25; Minneapolis, MN. NY: ACM; 2002; 415-422
- Belloni, L. (2007) *Psicomotricità in acqua Percorso educativo e terapeutico*. Erickson, Trento
- Bengtsson, PO. (2002) *Architecture-Level Modifiability Analysis*. ISBN: 91-7295-007-2, Blekinge Institute of Technology, Dissertation Series No 2002-2, 2002.
- Besio, S. (2003) They play and learn to play! First results of the Italian research project on play and children with motor impairment. 4th AAATE Conference, Dublin, Ireland. In: G. Craddock, L. McCormack, R. Reilly, H. Knops (eds.) *Assistive Technology – Shaping the Future*. Amsterdam, The Netherlands: IOS Press, 221-226.
- Bødker, S. (1991). *Through the interface—A human activity approach to user interface design*, Hillsdale, NJ: Erlbaum.
- Bondi, A.B., (2000) Characteristics of scalability and their impact on performance. *Proceedings of the 2nd international workshop on Software and performance*, Ottawa, Ontario, Canada, 2000, ISBN 1-58113-195-X, pages 195 - 203
- Bosch, J. (2000) *Design & Use of Software Architectures – Adopting and evolving a product-line approach*. ISBN: 0-201- 67494-7, Pearson Education, 2000.
- Brooks, A. (1999). *Virtual Interactive Space (V.I.S.) as a Movement Capture Interface Tool Giving Multimedia Feedback for Treatment and Analysis*. *Proceedings of the 13th International Congress of The World Confederation for Physical Therapy*, (p. 66). Yokohama, Japan.
- Brooks, A., Hasselblad, S., Camurri, A., Canagarajah, N., (2002) Interaction with shapes and sounds as a therapy for special needs and rehabilitation. In P. Sharkey, C. Sik Lányi & P. Standen (eds.) *4th Int. Conf. On Disability, Virtual Reality, and Associated Technologies*, 205-212. September 2002.

- Brooks, A. (2004a). SoundScapes. In . D. N. Snowdon, E. F. Churchill, & E. Frécon (Eds.), *Inhabited Information Spaces: Living with your Data* (pp. 89-99). Springer.
- Brooks, A. (2004b) SoundScapes: Multisensory Reciprocity through subliminal non-control. In R. Ascott (Ed.) *The 6th International Research Conference Consciousness Reframed: art and consciousness in the post-biological era* (pp. 37-44). Beijing, China.
- Brooks, A., Hasselblad, S. (2004c) CAREHERE – Creating Aesthetically Resonant Environments for the Handicapped, Elderly and Rehabilitation: Sweden. In: Sharkey, P., McCrindle, R., Brown, D. (eds.): *5th Int. Conf. On Disability, Virtual Reality, and Associated Technologies*, 191-198, September 2004.
- Brooks, A., Petersson, E. (2005). Recursive Reflection and Learning in Raw Data Video Analysis of Interactive 'Play' Environments for Special Needs Health Care. In *Proceedings of Healthcom2005. 7th International Workshop on Enterprise Networking and Computing in Healthcare Industry* (pp. 83--87). Busan, Korea.
- Brooks, T. (2006) SoundScapes - Beyond Interaction... in search of the ultimate human-centred interface. *16th International Conference on Artificial Reality and Telexistence (ICAT2006) : Advanced program*. Zhejiang, China, Zhejiang University of Technology, 2006. s. 17
- Brønsted, J., Hansen, K.M., Ingstrup, M. (2007a) A Survey of Service Composition Mechanisms in Ubiquitous Computing. *Workshop on Requirements and Solutions for Pervasive Software Infrastructures*.
- Brønsted, J., Grönvall, E., Fors, D., (2007b) "Palpability support demonstrated", In 'Proceedings of the 2007 IFIP International Conference on Embedded and Ubiquitous Computing (EUC 2007)'. Available at: <http://www.daimi.au.dk/%7Ejb/papers/bronsted07d.pdf>
- Bruner, J. (1992), *La Ricerca del Significato*, Torino, Bollati Boringhieri
- Büscher, M., Christensen, M., Hansen, K.M., Mogensen, P., Shapiro, D., (2007) Bottom-up, top-down? Connecting software architecture design with use. In Voß, A. and Hartswood, M. and Ho, K. and Procter, R. and Rouncefield, M. and Slack, R. and Büscher, M. "Configuring user-designer relations: Interdisciplinary perspectives", Springer Verlag, 2007. Available at: <http://www.ist-palcom.org/publications/files/Bottom-up-2.0.pdf>
- Carroll, J. M., (1995) *Scenario-Based Design. Envisioning Work and Technology in System Development*, John Wiley & Sons, Inc., New York.
- Carroll, J. M., (1999) Five Reasons for Scenario-Based Design. *Proceedings of the 32nd Hawaii International Conference on Systems Science*, Hawaii.
- Carroll, J.M. (2003) *HCI Models, Theories, and Frameworks: Toward a Multidisciplinary Science*. Morgan Kaufmann. San Francisco, CA

- Christensen, H. B. (2003) Using Software Architectures for Designing Distributed Embedded Systems. Technical Report CfPC-2003-PB-55, Center for Pervasive Computing, 2003.
- Clark, A., (1997) Being There: Putting Brain, Body and World Together Again. MIT Press: Cambridge, Massachusetts.
- Corry, A.V., Hansen, K.M., Svensson, D. (2006). Travelling architects – A new way of herding cats. Quality of Software Architectures (Lecture Notes in Computer Science 4214) Berlin: Springer, pp. 111-126.
- Dey, A. K., Abowd, G. D. (2000) Towards a Better Understanding of Context and Context Awareness,’’ Proceedings of the Workshop on the What, Who, Where, When and How of Context-Awareness, affiliated with the CHI 2000 Conference on Human Factors in Computer Systems, The Hague, Netherlands; ACM Press, New York, NY (2000), <ftp://ftp.cc.gatech.edu/pub/gvu/tr/1999/99-22.pdf>.
- Dey, A.K., Mankoff, J., Abowd, G.D., Carter, S.A. (2002) Distributed mediation of ambiguous context in aware environments UIST 2002, Fifteenth Annual Symposium on User Interface Software and Technology, CHI Letters 4(2), pp. 121-130, October 27-30, 2002.
- Dennett, D.C. (1987) The Intentional Stance, MIT Press
- DeRemer, F., Kron, H. H. (1976) Programming-in-the-Large versus Programming-in-the-Small. IEEE Transactions on Software Engineering, Vol. SE-2, No. 2, June 1976, pp. 80 - 86. Reprinted in Freeman and Wasserman, 1983, pp. 321 - 327.
- Diggins, T., Tolmie, P. (2003) The 'adequate' design of ethnographic outputs for practice: some explorations of the characteristics of design resources. Personal and Ubiquitous Computing 7(3-4): 147-158 (2003)
- Dourish, P., Bell, G. (2007) The Infrastructure of Experience and the Experience of Infrastructure: Meaning and Structure in Everyday Encounters with Space. Environment and Planning B: Planning and Design, 34, 3, 414-430.
- Ducatel, K., Bogdanowicz, M., Scapolo, F., Leitjen, J., Burgelman, J. C., (2001) Scenarios for Ambient Intelligence in 2010, IPTS Publications, EUR 19763 EN, 2001; available from: <http://www.jrc.es/publications/pub.cfm?id=587>
- Edwards, W.K., M.W. Newman, and J.Z. Sedivy, (2001) The Case for Recombinant Computing. Xerox Palo Alto Research Center Technical Report CSL-01-1, 2001.
- Edwards, W. K.; Newman, M.; Sedivy, J.; Smith, T.; Izadi, S. (2002a) Challenge: recombinant computing and the Speakeasy approach. 8th ACM International Conference on Mobile Computing and Networking (MobiCom 2002); 2002 September 23-28; Atlanta; GA. NY: ACM; 2002; 279-286.
- Edwards, W. K.; Newman, M.; Sedivy, J.; Smith, T.; Balfanz, D.; Smetters, D. K.; Wong, H. C.; Izadi, S. (2002b) Using Speakeasy for ad hoc peer-to-peer collaboration. ACM 2002 Conference on

Computer Supported Cooperative Work (CSCW 2002); 2002 November 16-20; New Orleans; LA; USA. NY: ACM Press; 2002; 256-265.

Edwards, W. K.; Bellotti, V.; Dey, A. K.; Newman, M. (2003) Stuck in the Middle: The challenges of user-centered design and evaluation for infrastructure. ACM Conference on Human Factors in Computing Systems (CHI 2003); 2003 April 5-10; Fort Lauderdale; FL. NY: ACM; 2003; 297-304

Edwards, W. K.; Newman, M.; Smith, T.; Sedivy, J.; Izadi, S. (2005) An extensible set-top box platform for home media applications. IEEE Transactions on Consumer Electronics. 2005 November; 51 (4): 1175-1181

Ehn, P., and Kyng, M., (1987) The collective resource approach to systems design. In Bjerknes, G., Ehn, P., and Kyng, M. (Eds.) Computers and democracy - a Scandinavian challenge. Aldershot, UK: Gower.

Ehn, P. 1988. Work-Oriented Design of Computer Artifacts. Stockholm: Arbeslivscentrum.

Eisenstein, J., Vanderdonck, J., Puerta, A. (2001) Applying Model-Based Techniques to the Development of UIs for Mobile Computers. Proceedings of the ACM Conference on Intelligent User Interfaces (IUI'2001), Albuquerque, NM; ACM Press, New York (2001), pp. 69–76.

Ellis, P., (1995) Sound Therapy. In Primary Music Today, issue 3, September 1995.

Ellis, P., Van Leeuwen, L. (2000) Living Sound: human interaction and children with autism. Paper presented at ISME commission on Music in Special Education, Music Therapy and Music Medicine, Regina, Canada, July 2000.

Ellis, P. (2004) Moving Sound. in MacLachlan, M. and Gallagher, P. (eds.) 'Enabling Technologies - body image and body function. Churchill Livingstone, 2004. Part 1, chapter 4, pp. 59-75

Emiliani, P. L., Stephanidis, C. (2005) Universal access to ambient intelligence environments: opportunities and challenges for people with disabilities. IBM Syst. J. 44, 3 (Aug. 2005), 605-619.

Engeström, Y. (1987). Learning by expanding: An activity-theoretical approach to developmental research. Helsinki, Finland: Orienta-Konsultit Oy.

Engeström, Y., & Middleton, D. (Eds.) (1996). Cognition and communication at work. Cambridge, UK: Cambridge University Press.

Floyd, C., (1984) A systematic look at prototyping. In R. Budde, K. Kuhlenkamp, L. Mathiassen, and H. Züllighoven, editors, Approaches to Prototyping, pages 1–18. Springer Verlag, 1984.

Gamelli I. (2001). Pedagogia del corpo, Roma: Meltemi.

- Gergely, G. and Watson, J. S. (1999). Early social-emotional development: Contingency perception and the social-biofeedback model. In P. Rochat (Ed.). *Early Socialization*. Mahwah, NJ: Lawrence Erlbaum Associates Inc.
- Gergely, G. (2002) The development of understanding self and agency. In *Blackwell's Handbook of Childhood Cognitive Development* (Goswami, U., ed.), pp. 26–46, Blackwell
- Ghizzioli, R., Rimassa, G., Greenwood, D., (2007) A Seamless Hybrid Communication System for Transient Locations. In *Designing for palpability Workshop at Pervasive 2007*, Toronto, Canada, May 13th-16th 2007. Available from:
http://www.ist-palcom.org/publications/files/Pervasive2007_ghizzioli_rimassa_greenwood.pdf
- Grønbaek, K., Kyng, M., Mogensen, P. (1997) Toward a Cooperative Experimental System Development Approach. In M. Kyng & L. Mathiassen (eds) *Computers and Design in Context*. MIT Press, Boston Massachusetts, pp. 201-238. 1997.
- Grönvall, E., Marti, P., Pollini, A., Rullo, A. (2006a) Active surfaces: a novel concept for end user composition, NordiCHI 2006, Oslo, Norway, 14-18 October, 2006.
- Grönvall, E., Pollini A., Rullo A., Svensson, D., (2006b) Designing game logics in dynamic Active Surfaces, MUIA06 at MobileChi 2006, September 12 2006, Espoo, Finland.
- Grönvall E., Piccini L., Pollini A., Rullo A., Andreoni G., (2007) Assemblies of heterogeneous technologies at the Neonatal Intensive Care Unit , Proceedings of the European Conference on Ambient Intelligence (AmI 2007), Darmstadt, Germany, 7-10 November 2007
- Halloran, J., Rogers, Y., Rodden, T., Taylor, I. (2003) Creating new user experiences to enhance collaboration. Proc. INTERACT 2003, Zurich, 479-486.
- Hasselblad, S., Petersson, E., Brooks, T., (2007) Empowered interaction through creativity. / *I: Digital Creativity*. 2007 ; vol. 18, nr. 2, June. s. 89-98
- Hornecker, E., (2005) A Design Theme for Tangible Interaction: Embodied Facilitation. In Proceedings of the 9th European Conference on Computer Supported Cooperative Work (E-CSCW'05) © Kluwer/Springer. pp 23-43
- Hornecker, E., (2006): Physicality in Tangible Interaction: Bodies and the world. Position Paper for the First International Workshop on Physicality, Lancaster University, 6-7 Feb. 2006, UK (organizers: Masitah Ghazali, Devina Ramduny-Ellis, Eva Hornecker, Alan Dix)
- Hornecker, E., (2007) Learning about Interactivity from Physical Toys. Accepted Full Paper for NORDES 2007 (Nordic Design Research Conference), Stockholm (online proceedings on www.nordes.org)
- Hornof, A.J., Cavender, A. (2005). EyeDraw: Enabling Children with Severe Motor Impairments to Draw with Their Eyes. SIGCHI Conference on Human factors in computing systems, Toronto, Canada. In: Proceedings of the SIGCHI/GI Conference on Human factors

- Houde, S., Hill, C. (1997) What do Prototypes Prototype? In Handbook of Human Computer Interaction, Amsterdam: Elsevier Science.
- Hunt, J.M. (1965) Intrinsic motivation and its role in psychological development. Nebraska symposium on motivation 13, 189–282 (1965)
- Hutchins, E. L., Hollan, J. D. and Norman, D. A. (1986). Direct Manipulation Interfaces. In D. A. Norman and S. W. Draper (Eds.). User Centered System Design, New Perspectives on Human-Computer Interaction, 87-124, Hillsdale, NJ: Lawrence Erlbaum.
- IONA Orbix/E Datasheet. <http://www.iona.com/whitepapers/orbix-e-DS.pdf>
- John, B. E.; Bass, L. (2001): Usability and software architecture. In Behaviour and Information Technology, 20 (5) pp. 329-338
- John, B. E., Bass, L. (2003) Avoiding "We can't change THAT!": Software architecture and usability. Tutorial materials presented at CHI 2003 (Ft. Lauderdale, FL, April 5-10, 2003).
- Jonsson, I.-M., Scott, N., Jackson, J. (1998) Disconnecting the Application from the Interaction Model. Proceedings of the Fourth ERCIM (European Research Consortium for Informatics and Mathematics) Workshop on User Interfaces for All, Stockholm, Sweden (1998), pp. 19–21, <http://www.ui4all.gr/UI4ALL-98/jonsson.pdf>.
- Jönsson, B., (Ed.) (2006) Design Side by Side. Lund, Sweden: Studentlitteratur. Available from: <http://www.certec.lth.se/doc/designsidebyside/>
- Kazman, R., Bass, L., Abowd, G., Webb, M. (1994) SAAM: A Method for Analyzing the Properties of Software Architectures. In proc. 16th International Conference of Software Engineering, pp. 81-90, 1994. 135
- Kazman, R., Barbacci, M., Klein, M., Carriere, S. J., Woods, S. G. (1999) Experience with Performing Architecture Tradeoff Analysis, In proc. of the 1999 International Conference on Software Engineering, pp. 54-63, 1999.
- Kazman, R., Bass, L. (2002) Making Architecture Reviews Work in the Real World, IEEE Software, vol. 19, pp. 67-73, 2002.
- Kazman, R., Bass, L., Klein, M., Lattanze, T., Northrop, L. (2005) A Basis for Analyzing Software Architecture Analysis Methods. Software Quality Journal, vol. 13(4), pp. 329-355, 2005.
- Kelly, M., Darrah, J., (2005) Aquatic exercise for children with cerebral palsy. Dev Med Child Neurol. 2005 Dec ;47 (12):838-842 16288676 (P,S,E,B)

- Kindberg, T., Barton, J., "A Web-Based Nomadic Computing System", *Computer Networks*, Elsevier, vol 35, no. 4, March 2001, and HP Labs Technical Report HPL-2000-110
<http://www.hpl.hp.com/techreports/2000/HPL-2000-110.pdf>
- Kirsh, D. (1995a). The intelligent use of space. *Journal of Artificial Intelligence*, 73(1-2), 31-68.
- Kirsh, D. (1995b). Complementary strategies: why we use our hands when we think. *Proceedings of 7th Annual Conference of the Cognitive Science Society*. Hillsdale, NJ: Lawrence Erlbaum.
- Kirsh, D., Maglio, P. (1994). On Distinguishing Epistemic from Pragmatic Action, *Cognitive Science*, 18, 513-549.
- Kyng, M., Kristensen, M., (2007) Supporting Palpability in Emergency Response, In *Designing for palpability Workshop at Pervasive 2007*, Toronto, Canada, May 13th-16th 2007 Toronto, Canada. Available from: http://www.ist-palcom.org/publications/files/Kristensen_Kyng_PervResp07_v1-0.pdf
- Le Boulch J. (1975) *Verso una scienza del movimento umano. Introduzione alla psicocinetica*. Tr. it. Armando, Roma 1978.
- Lapierre A. (2001) *Dalla psicomotricità relazionale all'analisi corporea della relazione*. Armando, Roma.
- Lund, H.H., Jessen C (2005) *Playware: intelligent technology for children's play*. Technical Report TR-2005-1, June, Maersk Institute, University of Southern Denmark.
- Machover, T. (2004) *Shaping minds musically*. *BT Technology Journal* ,Vol 22 No 4 . 171-179
- Marti, P., Lund, H. H., Bacigalupo, M., Giusti, L., Mennecozi, C., (2007). *Blending Senses: A Multi-sensory Environment for the Treatment of Dementia Affected Subjects*. In *Journal of Gerontechnology* 6(1).
- Malek, S., Seo, C., Medvidovic, N. (2006) *Tailoring an Architectural Middleware Platform to a Heterogeneous Embedded Environment*. " In *proceedings of the 6th International Workshop on Software Engineering and Middleware (SEM 2006)*, Portland, Oregon, November 2006.
- Markopoulos, P., Mavrommati, I., Kameas, A., (2004) *End-User Configuration of Ambient Intelligence Environments: Feasibility from a User Perspective*. *EUSAI 2004*: 243-254
- Medvidovic, N., Malek, S., Mikic-Rakic, M., (2003) *Software Architectures and Embedded Systems*. In *Proceedings of the Monterey Workshop on Software Engineering for Embedded Systems (SEES 2003)*, pages 65-71, Chicago, IL, September 24-26, 2003
- Medvidovic, N., Mikic-Rakic, M., Mehta, N., Malek, S. (2003) *Software Architectural Support for Handheld Computing*. *IEEE Computer*, Special Issue on Handheld Computing, vol. 36, no. 9, pages 66-73 (September 2003). Acceptance rate 5 of 87.

- Merleau-Ponty, M. (1962). *Phenomenology of Perception*. Colin Smith (translator). New York: Humanities Press.
- Mikic-Rakic, M., Medvidovic, N., (2002) Architecture-Level Support for Software Component Deployment in Resource Constrained Environments. Component Deployment, IFIP/ACM Working Conference, Berlin, Germany, June 20-21, 2002.
- Mikic-Rakic, M., Medvidovic, N. (2003) Adaptable Architectural Middleware for Programming-in-the-Small-and-Many. In Proceedings of the ACM/IFIP/USENIX International Middleware Conference (Middleware 2003), pages 162-181, Rio de Janeiro, Brazil, June 16-20, 2003
- Mascolo, C., Capra, L., Zachariadis, S., Emmerich, W., (2002) XMIDDLE: A Data-Sharing Middleware for Mobile Computing. In *Personal and Wireless Communications Journal* 21(1), Kluwer. April 2002.
- Mynatt, E. D., Melenhorst, A. S., Fisk, A. D., Rogers, W. A. (2004). Aware technologies for aging in place: Understanding user needs and attitudes. *IEEE Transactions on Pervasive Computing*, April-June, 36-41.
- Montessori, M. (1912) *The Montessori Method*. New York: F. A. Stokes.
- Nardi, B. (Ed.) (1996). *Context and consciousness. Activity theory and human computer interaction*, Cambridge, MA: MIT Press.
- Nardi, B.A., O'Day, V. (1999) *Information ecologies: using technology with heart*. MIT Press, Cambridge, Mass.
- Newman, M.; Sedivy, J.; Neuwirth, C. M.; Edwards, W. K.; Hong, J. I.; Izadi, S.; Marcelo, K.; Smith, T. (2002) Designing for serendipity. *Serious Reflection on Designing Interactive Systems (ACM SIGCHI DIS2002)*; 2002 June 25-28; London, England. NY: ACM; 2002; 147-156
- Norman, D. A. (1988) *The psychology of everyday things* New York: Basic Books.
- Norman, D.A. (1998) *The invisible computer*, MIT Press Cambridge, MA 1998
- Norman, D. A. (2004) *Emotional design: why we love (or hate) everyday things*, Basic Books, New York, NY.
- Norman, D. A. (2007). *The Design of Future Things*. New York: Basic Books.
- O'Brien, L., Bass, L., Merson, P., (2005) Quality Attributes and Service-Oriented Architectures. Technical Note. CMU/SEI-2005-TN-014. <http://www.sei.cmu.edu/pub/documents/05.reports/pdf/05tn014.pdf>
- O'Brien, L., Merson, P., and Bass, L. (2007) Quality Attributes for Service-Oriented Architectures. In Proceedings of the international Workshop on Systems Development in SOA Environments (May

20 - 26, 2007). International Conference on Software Engineering. IEEE Computer Society, Washington, DC, 3.

ObjectWeb Website - What's Middleware, Accessed: December 2007, <http://middleware.objectweb.org>

Overbeeke, C.J., Djajadiningrat, J.P., Hummels, C.C.M. and Wensveen, S.A.G., (2002) Beauty in usability: Forget about Easy to Use! In Pleasure with products: Beyond usability, Ed. Green, W.S. and Jordan, P.W., Taylor & Francis, London, pp. 8-18.

Overbeeke, C.J., Djajadiningrat, J.P., Wensveen SAG, Fren, J.W. (2001) Set Me Free, Give Me Degrees of Freedom. Proceedings of SSGRR2001 Conference. L'Aquila, 6 - 12 August. CD-rom.

PalCom Deliverable 12 (10.1) Initial Work Analysis Report. For internal dissemination only

PalCom Deliverable 16 (7.1) Multimedia Material. For internal dissemination only.

PalCom Deliverable 17 (7.2) Mock-up and prototype scenarios. For internal dissemination only.

PalCom Deliverable 18 (8.2) Prototypes. For internal dissemination only.

PalCom Deliverable 20 (1.1.1) Palpability: a Conceptual Framework for Palpable Computing. For internal dissemination only.

Palcom Deliverable 22 (2.3.1) Specification of Virtual Machine and Reference Implementation on selected Embedded Device(s). Description of use of VM in Application Prototypes. For internal dissemination only

PalCom Deliverable 24 (2.5.1) Design Issues for Resource Awareness and Management. For internal dissemination only.

PalCom Deliverable 32 (2.2.1): PalCom Open Architecture - First complete version of basic architecture. For internal dissemination only.

PalCom Deliverable 33 (2.13.1) Documentation of work analysis, participatory design and evaluation of exploratory prototypes. For internal dissemination only

PalCom Deliverable 37 (2.1.2): Palpability: Revised conceptual framework for palpable computing - Section I available from: <http://www.ist-palcom.org/publications/deliverables/Deliverable-37-%5B2.1.2%5D-palpability-revised-SectionI.pdf>

PalCom Deliverable 39 (2.2.2): Open architecture. Technical report, PalCom Project IST-002057, December 2006. Available from: <http://www.ist-palcom.org/publications/deliverables/Deliverable-39-%5B2.2.2%5D-Palcom-Open-Architecture.pdf>

PalCom Deliverable 50: Application prototypes final versions. Available from: <http://www.ist-palcom.org/publications/deliverables/Deliverable-50-%5B2.7.3%5D-prototypes-final.pdf>

- PalCom Deliverable 54. Open Architecture. Available from: <http://www.ist-palcom.org/publications/deliverables/Deliverable-54-%5B2.2.3%5D-open-architecture.pdf>
- Palcom Project Website, <http://www.ist-palcom.org/>
- Petersson, E. (2006) Non-formal Learning through Ludic Engagement within Interactive Environments. Malmö, Sweden : Malmö Högskola, 2006. 183 p
- Piaget, J. (1962). Play, Dreams and Imitation in Childhood. New York: Norton
- Pollini A., Grönvall E. (2006) Constructing assemblies for purposeful interactions. In Proceedings of Mobile Interaction in the Real World Workshop, MUIA06 at MobileHCI 2006, 8th International Conference on Human Computer Interaction with Mobile Devices and Services. 12 September 2006. Espoo, Finland.
- Pollini A., Grönvall E., Marti P., Rullo A. (2006) Constructing assemblies in the health care domain: two case studies. In Proceedings of Guide Mobili Virtuali 06, Virtuality 2006, 18 Ottobre 2006, Torino.
- Rellermeyer, J. S., Alonso, G. (2007) Concierge: A Service Platform for Resource-Constrained Devices. In: Proceedings of the ACM EuroSys 2007 Conference, Lisbon, Portugal, March 2007
- Renninger, K. A., Granott, N., (2005) The process of scaffolding in learning and development. In New Ideas in Psychology Volume 23, Issue 3, December 2005, Pages 111-114 Revisiting scaffolding: What do we think we know and what do we still need to figure out?
- Robertson, T. (2002) The Public Availability of Actions and Artefacts. Comput. Supported Coop. Work 11, 3 (Nov. 2002), 299-316
- Rochat, P., Morgan, R. (1995). Spatial determinants in the perception of self-produced leg movements in 3- to 5-month-old infants. Developmental Psychology, 31(4), 626-636.
- Rodden, T., Crabtree, A., Hemmings, T., Koleva, B., Humble, J., Åkesson, K.P., Hansson, P. (2004) Between the dazzle of a new building and its eventual corpse: assembling the ubiquitous home, Proceedings of the 2004 conference on Designing interactive systems: processes, practices, methods, and techniques, August 01-04, 2004, Cambridge, MA, USA
- Rogers, Y. (2006) Moving on from Weiser's vision of calm computing: engaging UbiComp experiences. In: P. Dourish and A. Friday (Eds.) Ubicomp 2006 Proceedings, LNCS 4206, pp. 404-421, Springer-Verlag
- Ross, D.A. (2004) Cyber Crumbs for successful aging with vision loss. Pervasive Computing, 3 (2004) 30-35
- Salzman, A. P., (1999) How to Define Aquatic Specialty Techniques: Operational Definitions. Available from: <http://www.aquaticnet.com/Article%20->

[%20How%20to%20define%20aquatic%20specially%20techniques.htm](#). Last accessed February 2008.

Savidis, A., Stephanidis, C. (2001) The Unified User Interface Software Architecture, in *User Interfaces for All—Concepts, Methods, and Tools*, Lawrence Erlbaum Associates, Mahwah, NJ, pp. 389–415.

Savidis, A., Stephanidis, C. (2004), “Unified User Interface Development: Software Engineering of Universally Accessible Interactions,” *Universal Access in the Information Society* 3, No. 3, 165–193.

Savidis, A., Stephanidis, C. (2005), “Distributed Interface Bits: Dynamic Dialogue Composition from Ambient Computing Resources,” in *Personal and Ubiquitous Computing*, Springer-Verlag, Berlin, Germany, <http://www.springerlink.com/index/10.1007/s00779-004-0327-2>.

Schmidt, A. (2002) *Ubiquitous Computing - Computing in Context*. Dissertation, November, 2002. Available from: http://www.comp.lancs.ac.uk/~albrecht/phd/Albrecht_Schmidt_PhD-Thesis_Ubiquitous-Computing_ebook1.pdf

Scholtz, J., Consolvo, S. (2004) Toward a Framework for Evaluating Ubiquitous Computing Applications. *IEEE Pervasive Computing Magazine*, Vol. 3, No. 2 (Apr-Jun 2004), pp. 82-8.

Schultz, U.P., Corry, E., Lund, K.V. (2005) *Virtual Machines for Ambient Computing: A Palpable Computing Perspective*. Position paper at the ECOOP'05 workshop on Object Technology for Ambient Intelligence (OT4AmI), July, 2005.

Shafer, S.A.N., Brumitt, B., Cadiz, J.J. (2001) *Interaction Issues in Context-Aware Interactive Environments*. Available from: <http://research.microsoft.com/easyliving/Documents/2001%2003%20Shafer.doc>

Shapiro, D., Mogensen, P. Buscher, M. (1996) Bricolage as software culture and practice, in *Proceedings of the COSTA4 Workshop on Software Cultures*, December, Technical University of Vienna.

Sharry, J., McDermott, M., Condron, J., (2003) “Relax to Win: Treating Children with Anxiety Problems with a Biofeedback Video Game” *Journal of the Irish Association for Counselling and Psychotherapy* v2 n25 p22-25 Sept 2003

Stringer, M., Fitzpatrick, G., Halloran, J., Hornecker, E. (2005) *Moving Beyond the Application: Design Challenges For Ubiquitous Computing*. Position paper presented at Aarhus 2005 workshop ‘Ambient Computing in a Critical, Quality of Life Perspective’, Aarhus, Denmark, 21st August.

Suchman, L. (2002) *Practice-based design of information systems: Notes from the hyperdeveloped World*. *The Information Society*, 18:1-6.

Suchman, L. (2007) *Human-Machine Reconfigurations*. Cambridge University Press, New York

- Svensson, D., Hedin, G., Magnusson, B. (2007) Pervasive applications through scripted assemblies of services", SEPS 2006, 1st International Workshop on Software Engineering of Pervasive Services, Lyon, France, June 2006. In Proceedings of IEEE International Conference on Pervasive Systems, 2007, p. 301-307. Available from: <http://www.ist-palcom.org/publications/files/seps06-scripted-assemblies.pdf>
- Tolmie, P., Pycock, J., Diggins, T., MacLean, A., Karsenty, A. (2002) Unremarkable computing. In Terveen, Loren (ed.) Proceedings of the ACM CHI 2002 Conference on Human Factors in Computing Systems Conference. April 20-25, 2002, Minneapolis, Minnesota. pp.399-406.
- Tran, Q., Calcaterra, G., Mynatt, E. (2005) Cook's Collage: Deja Vu Display for a Home Kitchen. In Proceedings of HOIT 2005, 15-32
- Vayer P (1971), Educazione psicomotoria nell'età scolastica. Tr. it. Armando, Roma 1974.
- Vayer, P., 1992, Per ritrovarsi nella complessità dei fenomeni umani. In: Perfetti, C., Pieroni, A., La logica dell'esercizio, ed. Idelson Liviana
- Valsiner, J. (1984). Construction of the zone of proximal development in adult-child joint action: The socialization of meals. In B. Rogoff, & J. V. Wertsch (Eds.), Children's learning in the "zone of proximal development" (pp. 65-76). San Francisco, CA: Jossey-Bass.
- Varela, F.J., Thompson, E. and Rosch, E. (1991), The Embodied Mind: Cognitive Science and Human Experience. Cambridge, MA: The MIT Press
- Vig, S. (2007). Young Children's Object Play: A Window on Development. "Journal of Developmental Physical Disabilities", 19, 201-215.
- Vygotsky, L. S. (1966). Play and its role in the mental development of the child. *Voprosy Psikhologii*, 12, 62-76.
- Vygotsky, L. S. (1978). *Mind and society*. Cambridge, MA: Harvard University Press
- Vygotsky, L. (1997). *Thought and language*. Massachusetts: The MIT Press.
- Waldo, J., (1999) The Jini Architecture for Network-centric Computing. *Communications of the ACM*, pages 76-82, July 1999. <http://citeseer.ist.psu.edu/waldo99jini.html>
- Wallon, H. (1959) *Psicologia ed educazione nel bambino*. La Nuova Italia, Firenze 1973.
- Watson, J. S. (1972). Smiling, cooing, and 'The Game.' *Merrill-Palmer Quarterly*, 18, 323-339.
- Watson, J. S. (1994). Detection of self: The perfect algorithm. In S. T. Parker, R. W Mitchell, and M. L. Boccia (Eds.) *Self-awareness in animals and humans: Developmental perspectives*. New York: Cambridge U. Press. Pp. 131-148.

- Weiser, M. (1991). The computer for the 21st Century. *Scientific American*, 265(3), 94-104.
- Weiser, M. (1993). Some Computer Science Issues in Ubiquitous Computing. *Communications of the ACM*, 36(7), 75-84.
- Weiser, M., Seely Brown, J. (1996) Designing Calm Technology, *PowerGrid Journal*, v 1.01, <http://powergrid.electricity.com> , July 1996. Also appeared as Chapter 6 - "The Coming Age of Calm Technology" in the book "Beyond Calculation - The Next Fifty Years of Computing" by Peter J. Denning and Robert M. Metcalfe, Copernicus/An Imprint of Springer-Verlag.
- Wejchert, J., (2000) The Disappearing Computer, Information Document, IST Call for proposals, European Commission, Future and Emerging Technologies., WWW: <http://www.disappearing-computer.net/mission.htm>
- WHO – World Health Organization (2007). International Classification of Functioning, Disability and Health Children and Youth Version. Available: www3.who.int/icf/onlinebrowser/icf.cfm?undefined&version=7.
- WHO – World Health Organization (2001). International Classification of Functioning, Disability and Health. Geneva, Switzerland: WHO.
- WHO – World Health Organization (1952). Constitution of the World Health Organization. World Health Organization: Handbook of basic documents. Geneva, Switzerland: WHO, 5, 3-20.
- Wood, D., Bruner, J.S., Ross, G. (1976) The role of tutoring and problem solving. *Journal of Child Psychology and Psychiatry*. Vol. 17, pp. 89-100