Department of Computer Science

Technical Report

DividingQuest: Using emotive interface personas in educational software

Sylvie Girard

Technical Report 2006-19 **December 2006**

ISSN 1740-9497

Contact Address:
Department of Computer Science
University of Bath
Bath, BA2 7AY
United Kingdom
URL: http://www.cs.bath.ac.uk

ISSN 1740-9497

DividingQuest: Using emotive interface personas in educational software

Sylvie Girard

MSc in Human Communication and Computing
September 2006

Department of Computer Science

University of Bath

Abstract

Research in educational technology has developed an interest in the capacity of software to be adaptable to the children's needs in terms of content, therefore enabling children to acquire learning at their own rate. Such smart teaching-aid systems aim to meet the challenge of providing instruction to students with a diverse range of abilities, interests and needs, increasing the child's learning motivation and self-confidence in their learning ability. Some researchers highlighted the benefits of using help features that are tailored to students' individual learning styles and skills to keep them motivated and increase their performance in using the software.

This report considers research on open-learner modelling, helping characters and emotions, and describes how this has been applied to the design of the "DividingQuest", a mathematical teaching-aid tutoring system for English year 6 children (10-11 years old).

The main research interest lies in the learner model holding information about teaching content relevant to the child's capabilities and abilities, as well as on the helping system offered to the child through the use of teaching characters with which the children might associate or look towards for help.

Designed using a participatory-design approach with year 6 children and teachers to adapt the software to children characteristics, the DividingQuest provides the foundation and necessary infrastructure for undertaking studies investigating the impacts of using emotive interface personas as helpers in Open-Learner Modelling tutors, on the appropriation and understanding of the user-model component by children.

Acknowledgements

I would like to thank my project supervisor, Hilary Johnson, for her on going support, advice and invaluable help throughout the project.

I would also like to thank Pascal Orzabal and Matt Cottrell for the time and effort they spent for the Parcitipatory-Design sessions as well as Constance Alleau, Claire et Patrick Girard for their precious evaluations of the system functionality.

Moreover, I would like to acknowledge the staff at Bathwick St Mary CE Primary School, Bath for assisting me in my project in the existing system evaluations with the 2005-2006 year 6 class.

At last, I would like to thank my family for their constant help and support.

Enclosed CD

The CD enclosed with this report contains:

- ⇨ A README document that describes how to run the system
- ⇨ The website implementation files
- ⇨ The database of the system in a sql file

Declaration

This dissertation is submitted to the University of Bath in accordance with the requirements of the degree of Master of Science in Human Communication and Computing in the Department of Computer Science. No portion of the work in this thesis has been submitted in support of an application for any other degree or qualification of this or any other university or institution of learning. Except where specifically acknowledged, it is the work of the author.

Author: *Sylvie Girard*

Signature:

Overview of the dissertation

I. Aims and Objectives

I.1 Introduction

Today's technologies are transforming people's interaction in their everyday activities, including the way our children live, learn, and play. It is therefore important to understand the impact the use of those new technological devices has on children to better support them in everything they do. Children do not want to be passively taught sitting in a classroom, they want to explore their environment, sharing their learning experiences with peers, parents or teachers. The learning process has therefore not only to be meaningful but also motivating.

Mathematics, along with English and Science, is a core curriculum subject. Studying this subject is essential to learn how to solve practical everyday problems, or to reason logically (*Hopkins 1996*). However, for a significant majority of primary school children, mathematics is not an appealing subject, considered as difficult to learn and teach due to its abstract nature. It indeed depends on a child grasping concepts that have no clear relation to anything they have encountered before (*Piaget, 1961; Copeland, 1979*).

To face this problem, many governments attempted to expand the 'arithmetic diet' through a variety of practical teaching methods. In 1999, the British government launched the DFEE's National Numeracy framework, introducing innovative teaching methods in the form of games, songs and themed lessons where children learn and practice the fundamentals of mathematics in concrete contexts that bring abstract concepts to life.

However, a major issue in the education system at this level is that a teacher cannot necessarily provide appropriate support to all the children, given that they can expect to manage a class of 25 or more. From their first year of education, children gain knowledge at different rates according to their abilities. Thus, teachers have to teach a variety of child the same subject, every child needing a personal amount of time to understand each concept in depth, in a system that does not enable individual tuition.

In order to overcome this problem research has been looking for a solution enabling the most able children to be stretched academically, whilst providing extra support at the basics for those who require it. As the cost of employing a person with the appropriate skill to undertake this one-to-one role is prohibitive, the direction chosen has been computer based tutoring systems, with its additional advantage of improving the ICT skill of the user while learning mathematics at their own rate.

A huge part of research in developing technologies for education within the classroom has been undertaken since the 1990s on building Intelligent Tutoring Systems, used to "develop a simulation of human problem solving in a domain in which the knowledge is decomposed into meaningful, humanlike components and deployed in a human manner" (*Polson & Richardson 1988*). A key element in the Tutoring Systems is the representation of the user model.

Research on user modelling undertaken by Bull and Kay (*1995; 2005; 2006*) highlighted the benefits of using "open learner modelling" in this area. Giving access for a learner to their own user model and even other students' models has indeed been found to improve learning skills and performance. Furthermore, teachers accessing their students' models have been able to modify the learning process and structure to their student's characteristics on account of the performance exhibited by the pupil evidenced by the model content, enhancing the efficiency of their teaching. However, the studies in this area have been undertaken on secondary children, and there is no trace of such results for primary school children.

At quite a different level, we are seeing the emergence of systems that model affective aspects such as emotion. There is a need to link this with the potential role of open learner models, and its issues: How can we help students to understand how their emotions and attitudes may affect their learning? Can this also help teachers interpret student achievement and interaction patterns? Whilst research in modelling emotions is considered as problematic, it is also known as modifying the user's perception of the software they are using (*Piccard, 1997*). In this context, can open learner models be beneficial for learners and what is the role of emotion?

According to Piaget's child development stage (*1961*) and the research studies of Chiasson, Druin, Gutwin and Inkpen (*Chiasson & Gutwin 2005; Druin 2004; Druin & Inkpen 2001*), designing for children has to be realised focussing on the children's needs and developmental stage. Children are not able to interact with the user models as adults would, and therefore their models have to be designed according to their characteristics. In the project, the interaction with their model will include the visualisation of their performance in the game and the customization of some system functionalities.

A common reason for failures in teaching are due to miscommunication from the teacher about the intended learning goals and what it means to achieve them as well as how to go about achieving them. Tutoring systems therefore have to take into account the teacher and the child's point of view on the learning process to produce motivating and interesting solutions.

As software designers are generally adults and, therefore, removed from the classroom environment, it can be difficult for them to understand exactly what motivates and amuses a child. In order to improve the efficiency and adoption of Tutoring Systems, new techniques have been realized to design and evaluate for children with children as Design Partners, and gave birth to a new area of Human-Computer Interaction, named Child-Computer Interaction.

Hence there is a large scope for incorporating the objectives of mathematics education within an Open-Learner Modelling tutoring system media whereby appropriate research into how to incorporate some kind of affect into the design of such tools may hold the key to render the learning and teaching of mathematics more enjoyable and motivating. The incorporation of emotional display into design might indeed enhance the understanding and the appropriation of the system by children and teachers.

I.2 Aim and Objectives

I.2.1 Overall Aim

The overall aim of the project is to provide software in the form of a Mathematical Teaching Help for year 6 children. This software will provide the foundation and necessary infrastructure for undertaking studies investigating the impacts of using emotive interface personas as helpers in Open-Learner Modelling tutors, on the appropriation and understanding of the user-model component by children.

I.2.2 Key Objectives

In order to achieve the project aim, the following objectives must be met:

- To research all relevant theoretical areas of the problem domain and thus gain a better understanding of the problem and its related topics.
- To investigate and understand the relevant human-computer interaction (HCI) and child-computer interaction (CCI) research that relates to designing software aimed at children, and to analyse how these issues can be taken into consideration when designing an Open-Learner Modelling Tutoring system for year 6 children.
- To review the known impacts found in the HCI research of using Affective Computing on learning and on the appropriation of computational devices.
- To compile a comprehensive list of criteria from the theoretical research and the investigation of the stakeholders of the project in order to elicit some requirements within the requirement analysis process.
- To critically evaluate current tutoring solutions and thus identify how children react to their knowledge representation, also assessing how their well-designed features can be applied to the design of the prototype solution.
- To undertake Participatory Design with year 6 children to investigate the best way to include emotions into the design of the interface persona of the tutor and to develop a highly interactive prototype solution according to those insights.
- To evaluate the prototype solution to ensure it meets the desired and intended functionality.

I.3 Project Outcomes

The overall outcome of the project is to provide a software architecture to be used as the main equipment basis for later experimental studies undertaken after the project for further research. It is aimed at evaluating the impacts of using emotive interface persona as helpers on learning achievements, child motivation and appropriation and understanding of the user-model component by children.

I.4 Overview of the chapters to come

Chapter II reviews the academic and theoretical research undertaken on the design of interactive computer-based mathematical learning tools.

Chapter III presents the naturalistic observations and investigations undertaken from the various stakeholders of the project in order to specify the requirements necessary for the development of open learner modelling tutoring system.

Chapter IV details the stages that were completed using a participatory design approach in order to establish complete set of design solutions corresponding to the requirements specified in chapter III.

Chapter V explains the implementation process, guided by the insights of the design process, that leads to the development of the high-fidelity prototype version of the mathematics helping tool.

Chapter VI outlines the evaluations undertaken on the prototype solution assessing whether it meets the desired and intended functionality defined in the requirements and design process.

Chapter VII concludes on the achievements of the project and provides a future research program including an empirical study using the software developed in the project to assess the effects of personas and personas presenting emotive affect.

II. Literature review

II.1 Introduction

The use of computers is becoming more frequent in every fields of human activity and touches people from every age and occupations. Due to this information revolution, it is essential that children have positive experiences with computers and technology to facilitate their interaction with technology in future life (*Wyeth & Wyeth 2001*).

Whilst Druin and Inkpen (*2001*) states that the use of computer can help supporting creativity and providing social experiences it is important to study the impacts the use of computer-based application can have on children's learning processes while they are taught mathematics in a classroom.

The intention of the literature review is to detail the research that has been carried out in order to understand all the different subjects and information that is directly related to this area and others that closely surround it.

The first section will take the reader through the child learning process, investigating the theories of child developments, the components of learning and how children learn mathematics.

The next section will then focus on the value of using computers as a medium of teaching, including the various teaching strategies and the design of Intelligent Tutoring Systems with a particular attention drawn on the impacts of opening the learner model and using pedagogical agents on learning.

Finally, the last section will present the field of Computer-Child Interaction along with its design and evaluation techniques for the design of children interactive products.

II.2 How children learn

Effective teaching lies in the ability of the teacher to set up a learning experience that makes the children willing to learn and help them grasp the concepts to integrate. In order for effective teaching to take place, each pupil must be engaged in the activity of learning. For this to happen, there is a need to understand the cognitive development of the learning process, enabling teachers and educators to tailor the specific abilities at each different stages.

In this section, we will first review existing child development theories and their implications on the learning process. We will then detail the key factors for learning to be successful, which consists in the apparition of an understanding and motivation. At last, we will outline how the mathematics differs from other subjects in its learning processes.

II.2.1 Child development theories

When considering how children learn, it is common to identify the group of children addressed by age. This is the grounding of child development theories, which state that children change with age according to five features of development; namely physical, social, emotional, intellectual and language (*Sylva and Lunt 1982*).

Early research in developmental psychology in children mainly investigated their limitations, assuming that children were restricted in their abilities compared with adults with respect to what they could comprehend and how they reasoned about situations.

Jean Piaget's theory of cognitive development (*1970*) classifies the key changes in intellectual and language development by providing a definition of a child in terms of age and cognitive skills. He claims that children 'naturally' pass through a sequence of developmental stages through which they attain conceptual knowledge by handling progressively more complex concepts in more complex ways. Table II.1 illustrates the stages of child development as defined by Piaget:

Stage	Ages	Key points for interactive product design
Sensorimotor	Birth to 2 Years	Children rely on what their senses perceive – they will be unable to interact with products. Develops an initial egocentric viewpoint that does not enable them to distinguish between themselves and the rest of the world.
Preconceptual thought	2 – 4 Years	Brief attention span, can only hold one thing in their memory at a time. Unable to read, but can understand simple instructions! Toward the end of the period, can build conceptual knowledge of mathematical adults and concepts.
Intuitive thought	4 – 7 Years	Children can use symbols and words. Children can distinguish reality from fantasy. In the latter part they can take into account the viewpoint of others. Will solve problems by empirical testing rather than logical reasoning. Dominated by their perceptions, are likely to be misled by what they see.
Concrete operations	7 – 11 Years	Children can classify things; understand the notion of reversibility and conservation. Can think logically but not abstract. The operations of "classification, ordering, construction of the idea of number, special and temporal operations" (Piaget 1964) are part of the children skills at that stage.
Formal operations	From 11 Years	Thinking is about ideas, they can consider various solutions without having to act them all out – can deal with hypothetical situations. It allows then to consider many different factors impacting on a given situation, using combinational understanding.

Table II.1: Piagetan stages of child development

Acuff and Reiher (*1997*) have developed Piaget's stages by describing the effect they can have on the design features for interactive systems. Their distinct groups illustrate the stages of child development in changing interests, humour, characters, contexts and settings. (*cf section IX.1.1*).

One of the main criticisms addressed to Piaget's theory is the rigidity of the stages, claiming that "children in the right context are able to develop certain understanding at a much earlier stage than he suggests" (*Kyriacou 1997, p30*) and that Piaget underplays the role of social factors in pupil's cognitive development.

Lev Vygotsky's theory of teaching and learning (*1987*) is focussed on the social aspects of learning, rejecting the idea that intelligence is fixed. Researching the impact of the aid of an instructor on the child learning process, he defined the concept of Zone of Proximal

Development. The research behind this concept is to analyze the potential of a child to rise about what they can achieve intellectually through collaboration with a teacher, making a transition from the current skills of a child to what it could achieve by imitating the instructor and finally do by itself. He concludes that children perform and develop better with help, and than tasks given to them should rather be testing their developing than their acquired learning.

The use of a tutor could help children from the concrete operational stage to understand and learn better. The presentation of abstract concepts in multiple ways could help them integrating new concepts and bridging the gap between the mathematical abstraction and the real-world problem application. It must also help them on their journey to understanding, maximizing the learning process in the zone of proximal development. An intelligent tutor containing an open-learner model could use the expert model and its understanding of the child knowledge to help him grasp new pieces of knowledge, reducing the help along with the comprehension of the notions by the child.

II.2.2 Understanding, human memory and motivation to learn

For a child to be able to learn a concept, it is essential that he understands the information, interprets and compares them to what he already knows and finally integrates the purpose and procedures of the learning content.

Norman (*1998*) explored how the human memory works to discover the best methods to support remembering and understanding. He defined three types of memory; these being memory for arbitrary things, memory for meaningful relationships and memory through explanations. Children use memory for arbitrary things, which consists in remembering without a deep understanding of the reasons for doing something in their learning of the alphabet with a song for example.

It is however argued that abstract associations or sequences are difficult to remember without constructing some kind of mental structure of the concept (*Norman 1998*). This is illustrated by the learning of the alphabet with a tune which rhythm and rhyme provide an appropriate structure that make is easier to remember.

In year 6, children will already possess some understanding of mathematical concepts but will be facing the learning of new abstract ones, not related to their previous experiences. It is therefore essential that learning has a sensible and meaningful structure to help children draw relationships between the items to remember and existing knowledge.

Quoting Richmond (*1970*), all new experiences must be related to experiences the child already understands. This theory of learning needed to be acquired by experience can be traced back to Confucius (450 BC.) who stated “Tell me, and I will forget. Show me, and I may remember. Involve me, and I will understand.” Children have to feel involve into the learning process, to be able to acquire links or connections to real world experiences and having them at their disposal to apply them in different contexts (*Hugues et al 2000*).

Hugues claims that the success of the understanding performance “depends on the familiarity of the element of the problem with associations already in the mind”, stating that the application of concepts in different contexts is fundamental for learning. This is applied by the National Numeracy Strategy (*DFEE 1999*) that promotes the expression of learning in real-world contexts for the knowledge to be assimilated efficiently and the teaching methods to be better understood.

Some finding from the studies of Margaret Donaldson (*1978*) showed that whilst children do not always interpret language as adults do, they are more likely to interpret the meaning of what they are told if they understand the context of the situation being discussed. Vygotsky and Liebeck (*1984*) support this theory by emphasizing the need for children to learn by having “authentic situations in which they must solve dilemmas” (*Vygotsky 1987*).

They advocate that learning metaphors and authentic examples can help to form the required connections between abstract concepts and the real world, by giving them a more “concrete focus” (*Donaldson 1978*).

A prime concern to all teachers is the motivation of the learner. Quoting Newton (*2001*), “ a press for understanding is likely to be accompanied by […] some provision that motivates the children to engage in these processes.” Yet, motivating a child to learn is still an issue.

Whilst the motivation sustain the learner’s attention and encourages him to motivate a child. Copeland (*1979*) explains that motivation does not only come from the child but also from the learning environment. He states that motivation is achieved if the experiences described in the learning session are closely matched to the child's ability to respond to them. A manipulation of the environment variables to expand upon existing knowledge and set of experiences might therefore result in an increase of motivation.

One way of motivating children to learn in the use of some kind of reward in case of success. In their experiment, Anderson et al (*1976*) demonstrated that verbal praise made the children spend more time practising the activities and therefore benefited the tutor and the learning process.

According to Keller (*1983*), there are four factors necessary to achieve optimal motivation: attention, relevance, confidence, and satisfaction. The use of a computer-based tutor could help achieving a good level of motivation in doing their learning activities. As the interactivity of the software could draw the child's attention to particular aspects of the concepts learned, it could also explore unfamiliar problems and show an evolution of the concepts according to difficult contexts. It should have some reference to the teaching activities realized in classroom and the curriculum expectations. The Intelligent Tutor could increase the level of confidence of the child by following Hammond's scaffolding theory of teaching (*2001*), increasing the difficulty of lessons to allow children to grasp more complex problems. At last, the use of colours, sound or animations and the opening of the learner model to the child could improve his satisfaction in using the tutor.

II.2.3 Learning mathematics

Mathematics is a core subject in the national curriculum, providing an intellectual challenge for children.

As Hopkins defined it in 1996, there is an inherent need for children to study mathematics at school, as it can help solving practical everyday problems, can be used in other area of learning and teaches you to reason logically.

Liebeck (*1984*) confirms this need by stating that "mathematics is useful for everyday life, for science, for commerce and for industry, because it provides a powerful concise and unambiguous means of communication and because it provides a means to explain and predict."

Mathematics investigates the manipulation of numbers, algebra, measures, shapes and the handling of data. It is typically associated with problem solving but also involves a certain amount of knowledge retention. The student is required to remember facts, learn and use newly acquired skills, understand conceptual structures and finally apply problem solving strategies and attitudes.

Year 6 pupils should be more than fluent in the mathematical principles and will be pushed to apply their knowledge when answering more abstract problems.

Teaching techniques, influenced by the national numeracy strategy aim to develop the child's communication skills and ability to use mathematical language along with the underpinning of 'sound' mathematical knowledge (*Hitcham, 2000*).

The key to developing mathematical knowledge is to form a base of understanding and to provide the means for applying it. Hopkins and al (*1996*) define three key skills that are needed for children to effectively tackle unfamiliar processes in mathematics: mathematical decision making, communication and reasoning mathematically.

The child could acquire communication skills by interacting with the computer-based tutor. The use of scenarios of application of knowledge could also help them to develop decision making skills and thus the ability to interlink mathematical concepts and deepen their understanding. In accordance with Bruner's findings (*1967*) the practice on a huge amount of relevant examples and scenarios to apply mathematical knowledge could accelerate the child's ability for mathematical reasoning, or at least build the basis of reasoning skills acquisition. At last, the use of animations, emotive characters, sound and colours could also help conveying ideas and concepts to children, helping them focus their thinking and practice using mathematical language appropriately.

II.3 Teaching Mathematics with computers

Walker (*1983*) stated that the use of computers provide more active learning, more vary sensory conceptual modes, less mental drudgery and are better tailored to individual learning rates. In this section, we will first study the needs and impacts of transferring teaching strategies to computer-based ones for the learning of mathematics. We then will review the research performed on intelligent-tutoring systems before to focus on their user model and its ability to be open to the learner or peers. At last, we will outline the possible outcomes of using animated pedagogical agents and affective computing in the design of computer assisted teaching systems.

II.3.1 Teaching Mathematics

Teaching is not just about textbooks, children must play an active part in learning. As Ecodomides states (*2000*), “the failure of so many instructional programs has been the result of an emphasis solely on content, with little regard for principles of instructional design to produce effective, efficient and appealing instruction”. Learning is an active, constructive, cumulative and self-regulated process in which the learner plays a critical role.

The DFEE (*1999*) defines ‘high-quality’ learning as oral, interactive and lively. Its achievement is not only an adoption of simplistic formula of drill or children learning from books by themselves. It invokes exposing children to appropriate material and concepts at pertinent times for their educational development to follow a suitable path. Teaching is therefore about support, consolidation and extension of knowledge (*Hammond 2001*).

In September 1999, the DFEE launched the national numeracy strategy, requesting all schools to provide a daily ‘numeracy hour’ to their students, during which the whole class must partake in oral, written and mental work to practise using their mathematical knowledge. The strategy, including a well-structured teaching framework and lesson ideas, led to widespread changes in the way mathematics is taught throughout England and Wales.

There are many approaches to teaching children, each of which aims to provoke different responses and nurture different modes of thinking within children.

Fisher (*1995*) identified some tools to be used by the teacher as teaching strategies, namely questioning children, creativity, social interaction, communication through group or pair work and coaching.

Questioning is an important part of all mathematics lessons and can be used either to challenge a child to recall information, or when building on previous knowledge to introduce new ideas. Whereas the former helps building confidence and maintain a consistent level of child participation in the lesson, the latter requires the child to think about new ideas and possibly apply their knowledge abstractly.

Another teaching method that plays a great role is group work. Involving children in group activities enables the teacher to explain concepts in different ways to aid clarity and resolve misconceptions, and to provide the children with the opportunity to learn form each other. This mode of teaching has been developed for computer-assisted learning through interface personas, providing instructions, feedback and a companion to the learner.

The DFEE promotes the use of computers in classroom due to their ability to provide a different medium for interaction and communication, and the willingness to learn it inspires. It enables to 'bring situations to life' such that children can apply their mathematical skills effectively, and without the problems associated with static materials. Computers can provide a break from writing in exercises books, working through textbooks and listening to the teacher present mathematical examples. Straker et al (*1989*) define three ways the computer can be used within the classroom: as a teacher aid, a learning resource or a tool for the child to use in doing mathematical tasks.

With interactive resources feeding children's natural curiosity and enthusiasm for interaction, the introduction of computers in the mathematical teaching techniques is a valuable addition and an effective way for the teacher to capture the attention of the child. There is still a real need for tutors to use cognitive mapping for the student to directly recognize when material is related. It should also provide students with questions to demonstrate the use of concepts in different contexts. The tutor finally must be a coach, allowing the child to ask questions, summarizing the material that has been learnt and giving the child feedback on its progress.

II.3.2 Intelligent tutoring systems

Providing a personal training assistant for each learner is beyond the budget for most organisations, including schools in our scenario. However, a 'virtual' training assistant that captures the subject matter and teaching expertise of experienced trainers provides a new option. The goal of ITS is to provide the benefits of one-to-one instruction automatically and cost effectively.

Research in the field of computer-assisted instruction (CAI) has been described under numerous names but all refer to computer software that aims at instructing. ITS and adaptive hypermedia systems add the "intelligence" to it.

Polson& Richardson (*1988*) defined the aim of an ITS as to "develop a simulation of human problem solving in a domain in which the knowledge is decomposed into meaningful, humanlike components and deployed in a human manner".

As Ong and Ramachandram (*2000*) explained "Unlike other computer-based training technologies, ITS systems assess each learner's actions within these interactive environments and develop a model of their knowledge, skills and expertise. Based on the learner model, ITSs tailor instructional strategies, in terms of both the content and style, and provide explanations, hints, examples, demonstrations and practice problems as needed."

From the apparition of the first ITS three decades ago, a number of tutoring systems have been based on Anderson's Adaptive Control of Thought (ACT*) theory of learning (*Anderson, 1995*). ACT* sees the student's problem-solving behaviour as being goal-oriented, not as behavioural response to stimuli. Those systems looked at the generalization as an automatic process, which takes place as students are studying examples.

Fifteen years ago, a 'second generation' of computer tutors began to appear, due to the advances in Artificial Intelligence, educational and cognitive theories on learning. These technological developments helped building intelligent tutors. Whilst they seem to be effective in helping learners, they do not provide the same kind of individualized attention that student would receive from a human tutor.

The third generation of tutors seems to focus on the effectiveness of human tutors. A significant effort is given to bridge the gap between the student and the tutor in evaluating and improving the different models of the ITS.

Although this approach enables the development of powerful tutoring systems, developing an expert system that provides comprehensive coverage of the subject material is difficult and expensive. For the purpose of the project, we will focus on the user and instructor model, using an expert system with restricted coverage of the subject material.

There need to be a section about the difficulty of having a system understand users learning and tutor output accordingly. The tailorability comes from utilising a student or user model within to system

II.3.3 Open User Modelling

Part of the user modelling research community studies user models considering the user as a learner and using the models in the design of educational software and ITS. The learner model is the foundation of system with the potential to treat learners as individuals. Research in user-modelling for computer-assisted instruction systems gives an increasing interest in the open learner models (OLM), models of the user that are available for viewing by the learner and sometimes others.

OLM offer the potential to help learners and teachers to reflect on their own knowledge, misconceptions and learning processes. It can help motivating students who aspire to perform at the highest standard as well as giving them ideas of their own performances against average results. There is also the rather interesting case of the student who wants to learn precisely what is essential to pass and not more.

OLM can also serve as a focus for mixed-initiative interaction where the learner can be challenged about their learning and where the learner can also challenge the system about their learner model.

Learner models open to instructors can be used to allow the instructor to follow the evolution of a student's knowledge (*Rueda et al., 2003*); to help instructors adapt their teaching to the individual or the group (*Grigoriadou et al., 2001; Zapata-Rivera & Greer, 2001; Yacef, 2005*); to help instructors organise learning groups (*Mühlenbrock et al., 1998*);

or to allow the teacher to combine the learner data with other information about the learner, obtained from outside the system (*Jean-Daubias & Eyssautier-Bavay, 2005*).

Learner model information can also be provided to parents, to enable them to follow their child's progress (*Zapata-Rivera et al., 2005*).

A big issue in research is to incorporate OLM into conventional learning systems. Due to their large collection of data about individuals, new ways of making this data more useful as detailed models of learner development have to be realized. By working closely with classroom teachers there is also a need for defining interfaces where teachers and learners could see the learner models.

Bull and Kay (*2005, 2006*) proposed a method to describe and analyse OLM: The SMILI☺ Open Learner Modelling Framework. The framework aims to enable comparisons of features of OLMs in different systems by categorizing them according to seven purposes of user-model openness.

The externalisation of the learner model has been realized through various forms and contents, such as skill meters showing learner progess (*Weber & brusilovsky 2001*), Bayesian network graphs (*Zapata-Rivera & Greer 2001*), hierarchical tree structures (*Kay 1997*), or textual descriptions of knowledge and misconceptions (*Bull & Pain 1995*).

Open learner models for children need to be simple in format in order that the children can understand them. Examples include knowledge level represented as coloured magic wonds (*Bull et al, 2005*) or smiling faces to represent the different levels of knowledge (*Bull & McKay, 2005*).

Some research recently evaluated to what extend and in which circumstances students would use the view of their learner models. Whilst some studies (*Bull&Kay 2005*) showed positive results on groups of 8-9 years children, it also outlined some cases where the presence of the OLM had a negative effect on the children learning process. The choice and content of opening the learner model therefore would have to be made carefully.

II.3.4 Pedagogical Agents and emotions

Researching the role of affect and emotion within the context of computing is an area that has shown an emerging prominence in the last 10 years (*Piccard, 1997*). These capabilities have moved along quite quickly to the point where information technology not only affords a "social presence" but can now create "affective presence" as users interact with computers.

A new breed of intelligent learning environments called learning companion systems was developed over a decade ago. In contrast to an intelligent tutoring system, in which a computer mimics an intelligent tutor, the learning companion system assumes two roles, one as an intelligent tutor and another as a learning companion.

This was the beginning of including animated agents into ITS, with the birth of the term pedagogical agents. Those pedagogical agents are a real potential to motivate and engage the user in the learning experience (*Gulz,2006*).

Whereas a classing intelligent tutoring system is invisible and fairly abstract, the addition of an animated pedagogical agent to the interface provides elements of embodiment, visibility and personality; in addition to the ability to communicate in an intelligent manner, a pedagogical agent should, according to Lester et al. (*2000*), have socio-emotive abilities and be lifelike. Consequently, the addition of animated agents to ITS opens up the possibility for learners to have a personal relationship and an emotional connection with the agent, which in turn may promote interest in the learning task (*Moreno et al. 2001*).

The pedagogical agents can have different roles. Some of them are teacher, instruction or coach (*Oivatt & Adams, 2000*), learning companions (*Rickel and Johnson 2000*), whereas others are guide or presenters (*Norma & Badler, 1997; Beskow &McGlasham, 1997*). A last category of pedagogical agents are the PPP persona presentation team (*André et al, 1998*).

Whilst the potential of animated agents to make the experience more engaging is well established (*Walker et al, 1994; Takeuchi & Naito, 1995; Lester et al, 1997; André et al, 1998*) there is also a risk of the user being distracted by it with bad effects on learning (*Rickenberg and Reeves 2000; Moreno et al, 2001*).

Several studies illustrated the fact that users react differently to an animated agent based on their own personality.

Shneiderman (*1998*) argues that interface personas can make users feel inferior if their mode of instruction is too patronising. Thus, the pedagogical agents should not be intrusive to the functionality of the system, and its affect on the usability of the system as a whole must be carefully planned and considered.

II.4 Designing with and for children

Interactive products for children regroup three main categories: education, entertainment and enabling products such as web browsers or word processors for children. Special approaches are needed throughout the lifecycle when the intended users are children which lead to the research in methods for designing and evaluating such products with and for children.

In this section, we will first describe the issues of interaction and effectiveness of products for children which are the main concern of the field of Child-Computer Interaction (CCI). We will then concentrate on how to create educational software for and with children, prior to report the different methods used to evaluate children products.

II.4.1 From Human-Computer Interaction to Child-Computer Interaction

Over recent years there has been a significant increase in the published work relating to children and interaction design. CCI is emerging as a vibrant sub-field of Human Computer Interaction, following pioneering work by Rogers and Scaife, Druin and Kafai.

When designing for children, general HCI interface design rules, guidelines and principles can be applied (*Schneiderman 1998; Dix et al 2004; Preece et al 2002*). Nevertheless some features of children, different from adults need to be taken into account when designing software for them.

Gilutz and Nielsen (*2000*) reported for example a large number of cases in which their usability findings for 8-12 years old children highly differed from results generally obtained with adults. Some adaptations have therefore to be made like the words used in heuristics (*McFarlane&Pasiali 2005*) which led to the creation of new guidelines (*Druin, 1999*), principles (*Chiasson & Gutwin 2005*), design and usability heuristics (*Hanna, Risden et al, 1997; Nielsen 2001*).

The problem in using those evaluation techniques is that children will not necessarily fully understand the implications of problems in the interface, due to their lack of experience of interfaces (*McFarlane et al, 2005*).

In the case of a tutoring system for children, it is vitally important that the mode of interaction between the child and computer is efficient and intuitive so that children can access the functionality of the system and focus on the task without the need for support or instruction.

Preece et al (*2002*)'s claims that "…aesthetics of an interface can have a positive effect on people's perception of the system's usability" and thus a large part of design has to concern the interface in the system.

The interaction between the user and the computer can be seen as a dialogue of communication. It is important that the interaction style is appropriate for the task domain and dialogue type, so that the communication is intuitive and allows the user to access the real functionality of the system. Newman et al (*1995*) recognise 8 interaction styles, organised into 3 categories: key modal, linguistic and graphical direct manipulation.

Druin (*1999*) concludes that children aged 5 – 7 want interfaces that they can easily control; they want interfaces that 'respect' them, those being interfaces that are not too simple. When designing for children, the interaction style must suit to the children level of computer literacy. Graphical direct manipulation therefore appears the most appropriate interaction style given the problem domain. It is a highly interactive means of human-computer interaction and is advantageous as it involves the user constantly providing instruction and receiving feedback on the status of those instructions. With the use of appropriately labelled buttons and symbols, point-and-click interfaces allow easy navigation through the system as the user should always be aware of "where they are going". This is highly relevant for child users who have limited experience of computer use and thus require additional clues on how to achieve their tasks.

Designers of children's technology must concentrate not only on the mechanics of their interfaces but also on features that will keep children engaged (*Chiasson & Gutwin*

2005). Icons can be used to convey meaning as they can present a pictorial clue of the action performed by clicking on them. This allows children to rely on their visual sense of perception and thus make inferences about the action that an icon represents. Work by Hanna et al, (*1999*) suggests that the difficulties that children have with reading can be reduced by the use of 'visually meaningful icons', 'thoughtful cursor design', and the addition of features such as rollover, audio, animation, and highlighting. There is some specific research on the selection of icons for children's interfaces. Uden and Dix (*2000*) and Baecker, Small and Mandler (*1991*) report that children prefer animated icons, and it is often the case that such icons can offer more information than those with static representations.

II.4.2 Children as Design Partners

Designers in the CCI field advocates that involving children in different phases of the design process is vital to ensure that most of the requirements needed are included (*Mazzone & Read 2005*).

Alisson Druin (*1999*) introduced a model of the roles of children in the design process, illustrated in Figure 1:

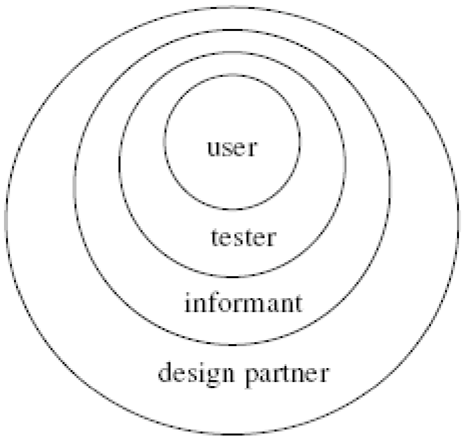

Figure 1.1: Roles children can play in the design process

Research has been carried out in this area to explore the suitability of design techniques for involving children in specific phases of design.

When children are Design Partners, they work as equal members of the design team, helping by identifying problems and solutions to improve the technology they may need in support of their activities (*Mazzone & Read 2005*).

One research approach to designing with children is outlined by Druin (*1999*). Cooperative Inquiry takes elements from both Contextual Inquiry and from Participatory Design and adapts them for use with children, to investigate changing the way children are consulted regarding the development of software that affects them. Druin found from her findings children in the 'concrete operational stage' to be the age group the most effective when prototyping, as they are "verbal and self-reflective enough to discuss what they are thinking…[not] too heavily burdened with pre-conceived notions of the way things 'are supposed to be'." It is important that adults and children work together on the design, as no one partner should be more important than another, even though adults may assume the child-like appearance of the materials implies otherwise. One point to note is that the design materials need to be carefully chosen to match the activity, as one standard set of equipment is unlikely to be suitable for all ideas.

However it is important to bear in mind that children are not "little adults" and that care must be taken when planning design sessions to ensure that the activity is set at the right level to enable the children to be able to express their ideas in a useful format.

Using participatory design with children has proven to be a useful technique along with cooperative inquiry and helps to design systems more effective and appealing for the future users.

II.4.3 Evaluating systems with and for children

The methods used to evaluate a product are determined by the stage of development of the product at the point of evaluation, the availability of resources (including users and design experts) and the feature(s) of the product that are being evaluated.

Evaluators methods in the CCI field contains HCI general methods adapted to children (Heuristic Evaluation, Cognitive Walkthroughs, Structured Walkthroughs, Think Aloud, Surveys) and some methods especially designed for children (Co-discovery, Peer-tutoring). They can be separated into two groups, evaluating either by observing or asking children.

In observational work with children, the age, the position, and the appearance of the observer may influence the findings (*Hanna, Risden et al. 1997*). Some general characteristics of children that need to be taken into account when carrying out evaluations are that they have developing capacity to verbalise, they have different levels of extroversion, their knowledge and skills may be different, and although generally being very honest in their judgements, the reliability of reported data is questionable, (*Druin, 1999 ; Markopoulos and Bekker 2003*).

Rode et al (*2003*) highlight that any usability test must fit within the school sessions; that it has to fit the framework of the school and that it should not cause 'lesson stoppers'. Rode et al. (*2003*) use this term to describe events or materials like games, animations, technology failures, etc., that distract children from curriculum objectives and could derail a lesson.

Evaluating products by asking children can be realized in an informal way about products that they have seen either alone or in groups. They can also be asked in a more formal way using interviews or questionnaires. Children aged 8 – 11 are easier to survey and can complete simple questionnaires; however, they also tend to be very literal and cannot easily understand negatively constructed questions. Whereas practice with adults favours interspersing positive and negative statements when determining attitudes, this is not advisable with children. (*Mazzone & Read 2005*). Donker and Markopoulos (*2005*) found that the think-aloud technique originally developed for adults could work well also for children between ages 8-14 and helped to uncover more problems than post-talk interviews and written questions.

Children have their own abilities, curiosities, needs, skills and expectations. Their goals while using computer are education or entertainment rather than a common adult goal of productivity (*Chiasson & Gutwin 2005*). New criteria have to be taken into account when evaluating product for them such as fun.

Schneiderman (*2004*) states that "Fun-filled experiences are playful and liberating – they make you smile. They are a break from the ordinary and bring satisfying feelings of pleasure for body and mind". Including a notion of "fun" into the design process can increase the motivation of the child to use the software, which leads to the inclusion of fun features in the design such as "alluring metaphors, compelling content, attractive graphics, appealing animations and satisfying sounds".

Many researchers have explored the relationship between fun, play and learning, reasoning that fun contributes to being motivated to pursue an activity, and as such can also contribute to learning effectively (*Malone & Lepper, 1987; Prensky, 2000*).

II.5 Conclusion

The study of several areas relevant to the problem domain raised some interesting issues that form a theoretical base to the project. We will know highlight the most relevant ones for the effective design of computer-based tutors to help children in their learning process.

The structure and content of the tutor must be designed to maximise the potential academic development of each child following the child skills inferred by their different stages of development. Tutors provide an ideal environment for structured questioning, and can be used following Hitchman's scaffolding theory of teaching (*2000*) to help children increasingly grasp new concepts and consolidate their knowledge. This also supports Vygotsky's theory of zone of proximal development by the use of examples and understanding of misconceptions with the ITS, with a possible acceleration of the child development.

More computer-based systems help the learning process by providing the basis for manipulation and exchange of information in a different form than the traditional teaching methods. The interactivity of the system allows provides a visual perspective on abstract concepts and allows children to rely on their sense of visual perception. Using metaphors, animations and graphical information, it can help to bridge the gap between the abstract mathematical problems and the real-world situation of their application (*Vygotsky & Liebeck*

1984; Donaldson 1978). The use of pedagogical agents may help constructing those associations and have a clear understanding on the child level and achievements.

A key consideration when designing for learning is the motivation the teaching technique provide to the child. They must have the desire to make a positive mental effort towards the learning task. Keller (*1983*) states that motivation is function of four key factors, which can clearly be integrated into the design of computer-based tutors. Attention can be drawn on the work by using pedagogical agents and graphical animations, still being careful for the tool used not to be a distraction to the learning. A design closely realized with the help of teachers can insure a good reference level to the concepts and decision-making strategies children are used to manipulate while learning mathematics at school. As stated before, the exploration of concepts through multiple view and in different contexts could help children gain some confidence in their learning and make the necessary connections between abstract concepts and real-world. At last, the OLM offers the child an understanding of his own level and could help his satisfaction in the learning process through control and greater freedom and flexibility.

The research in ITS is currently focussing on the improvements of the learner model to make the understanding between the user and the tutor more effective. Bull et al (*2005*) demonstrated that children from the 'concrete operational stage' can understand and use an OLM in their learning and that this may be beneficial for their performances. The inclusion of pedagogical agents into the ITS have proven to change the way children use the ITS and some findings indicate that this could increase the motivation of the child to use the application and its performance while using the system. Studies in OLM for children (*Bull & Kay 2005*) using smiling faces as representation of knowledge seemed to help the motivation of the children to use the software but the impacts on the child understanding of its own performances or on the learning process are not actually well defined.

From the research into CCI, we drew out some important considerations to be taken into account while designing for children. This knowledge should be applied throughout the design stage using children as design partners with the aim of producing a highly learnable system so that children can easily apply their skill base without having to overcome problems with the system. An evaluation of existing solutions following the same criteria could also help designing a tutor better suited for the learning of mathematics for year 6 children.

The next stage of the project therefore should be empirical studies and investigations into relevant practical sources to examine how the tutor will be integrated into the lessons and the role it could play. Information gathered by stakeholders and evaluation of existing solutions will provide insights to build the tutor in an effective way, eliciting design considerations. The next section will review the information gathered from these sources when coupled with the theoretical base established in this chapter to produce elicited design requirements for the open learner modelling tutoring system.

III. Requirements

III.1 Introduction

The literature review has led us to explore the academic and theoretical bases both justifying the need for and underpinning the design of an interactive computer-based mathematical learning tool for use on year 6 children. However, the requirement analysis process has to be balanced between its theoretical base, the practical issues surrounding the tutor stakeholders and the evaluation of existing tutoring solutions. Through the empirical studies undertaken, we therefore will be able to appreciate the structure and content of the learning activities along with the rationales of those learning processes. This will help us gain an insight into the conditions under which the tutor will be used and the constraints on its different components, in an attempt to maximize the potential of using ICT for mathematical learning in schools.

This process will lead to a detailed understanding of the children's needs in their working context along with their requirements and aspirations, so that the product designed can better support their goals (*Preece et al, 2002*). It will enable us to produce a stable set of requirements that will form the basis for the design of the prototype solution and provide a guideline framework for evaluating the system upon its completion.

In this section, we will first detail the sources that have been chosen for the establishment of the requirements document and the structure of this document. We then will review the different requirement elicitation activities as well as the validation of the requirements.

III.2 Specifying the requirements

III.2.1 Sources of requirements

The requirement elicitation process is inherently generated through a detailed study of the problem domain. To ensure sufficient depth of the analysis a wide spectrum of sources have been chosen within the domain, illustrated in figure III.1.

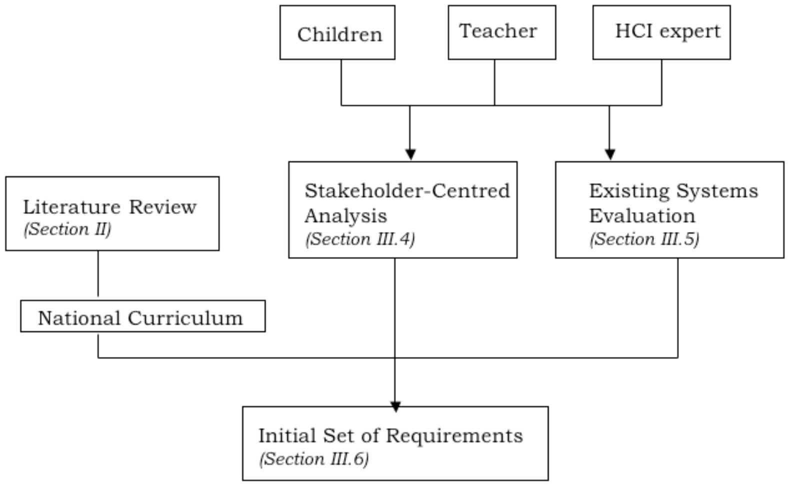

Figure III.1 Sources of requirement elicitation process

As illustrated in figure III.1, the elicitation process has been completed using theoretical and empirical evidence, in line with the framework introduced by Druin (1999). The process indeed includes various activities such as requirement gathering from the literature review, teaching and learning observations, interviewing and questioning, results taken from participatory-design sessions and an evaluation existing solutions.

As previously discussed, the literature review provides the theoretical basis for the project domain in generating requirements built on successes and failures of past academic work. It also guides us along the processes of stakeholder-centred analysis and existing tutor evaluation in providing support and background research.

Stakeholder-centred requirement analysis aims to understand the stakeholders' goals, activities and practises in order for the final system to support more efficiently their tasks and expectations.

The stakeholder-centred requirement analysis involves the following sources:

- ⇨ Children between the ages of 10-12 from the 2005-2006 year 6 class at Bathwick St Mary CE Primary School, Bath;
- ⇨ Mr Matt Cottrell, year 6 teacher for the year 2005-2006 at Bathwick St Mary CE Primary School, Bath;
- ⇨ Pascal Orzabal, 10 years old year 6 participatory-design child throughout the design process.

A further activity is the evaluation of existing solutions that gives an understanding of the products currently on the market. An analysis of their major failures and successes with a particular interest given to their user-modelling and representation.

As shown in figure III.1, an initial set of requirements can be elicited from the previous activities. It is however necessary to consider Carroll's task-artefact cycle (*1991*) in the requirement process. As Carroll states, the set of requirements elicited from empirical studies not only represents the tasks users *currently* perform but also the ones they *want* to perform. When designing for a new product, new requirements can be inferred from users as they may require support in additional activities, or their needs may suffer from huge modifications.

A last activity of the analysis process therefore is the validation of the initial sets of requirements elicited to both support Carroll's task-artefact cycle and ensure that the requirements represent the key issues and goals that need to be supported by the final system. This validation process, performed with the help of Matt Cottrell, is detailed in section III.6.

The next section will define the structure in which the elicited requirements will be gathered to form the sets of requirements that led to the design process.

III.2.2 Structure of requirements

The requirements will be structured according to the following framework, based on Sommerville (*2001*) and Preece et al (*2002*) requirement classifications. While requirements are traditionally classified as either functional or non-functional, Preece et al (*2002*) state that this classification does not suit the requirement specification of interactive systems, due to the broadness of the two categories involved, and therefore needs to be represented by a larger set of categories.

The requirements gathered from each requirement elicitation activity will satisfy the following structure:

⇨ ***Functional Requirements***: requirements describing the functionality or services the system is expected to provide.

⇨ ***Non-Functional Requirements***: requirements not directly related to the specific functions of the system. They may define constraints or performance of the system as well as facts and assumptions relevant to the system.

⇨ ***Usability and user-interface Requirements***: requirements capturing the usability goals and association measures of the system.

⇨ ***User Requirements***: requirements representing the characteristics, abilities and skills of the intended user group.

The following sections will detail the requirements gained from the analysis and the processes that were undertaken in order to elicit them.

A comprehensive list of requirements conforming to this structure will be generated from the eliciting activities and can be seen in section IX.2.5 in the Requirements Appendixes.

III.3 Requirements gathered from the Literature Review

The literature review has led us to gather requirements from a broad perspective within the problem space. Through the work of Piaget and Vygotsky it highlighted how children learn and develop. Donaldson(1978) and the national curriculum introduced different teaching strategies and the way children develop expertise.

Problem solving and the effectiveness of example-based learning have also been overviewed through the research of Anderson, Newell and Simon. The work of Alison Druin and the ChiCI research group along with HCI research raised issues involving learnability, flexibility and robustness of the system and the importance of the user-interface while designing for children.

The requirements sourced from the literature review were included in the set of initial requirements in section III.6. Some of them have been adapted to the needs and expectations of the stakeholders of the system during the stakeholder-centred analysis, as defined in the next section.

III.4 Stakeholder-Centred Requirement Analysis

A Stakeholder-Centred requirements analysis was undertaken for the project, analysing the stakeholders' goals, activities and practises. The analysis included naturalistic observations of children using the school computers and interviews of teachers on their needs and expectations concerning the software to be developed.

This is critical in an educational-purposed software, as designers must aim to support the learners' needs in consideration of their computing and mathematical skills. The tutoring system therefore must ensure learning and teaching consistency between the various teaching tools in the classroom. It must also fit in with the teacher's lesson structure to limit misconception and confusion for children. See section IX.2.2 for a full description of the observations and interviews undertaken.

The next requirement elicitation activity was the evaluation of existing system, described in the next section.

III.5 Requirements gathered from evaluation of existing systems

By analysing appropriate mathematical tutoring systems, we aim to uncover the advantages and well-designed features of the interface, interaction style, tutorial content, user-model and representation of affect components of the systems that could be carried forward into the design stage. The process will also highlight the design failures or poor-design features of the existing products and their possible improvements to be reused in the design of the mathematical tutor.

A representative set of current web-based tutoring systems has been selected, based on the tutors used frequently at school by Pascal Orzabal as well as an appropriate tutoring system presenting a user model with emotions designed for year 3–4 children. These are:

⇨ BBC Dance Mat Typing
http://www.bbc.co.uk/schools/typing/

Whist the content of the tutor is not mathematics, it has been chosen since it provides high quality assessment material from children to be addressed on their use of the keyboard and mouse and on their perception of the usefulness of interface personas in tutoring systems.

⇨ BBC website Key Stage 2 Division-themed games
http://www.bbc.co.uk/skillwise/numbers/wholenumbers/division/mental/game.shtml
http://www.bbc.co.uk/skillwise/numbers/wholenumbers/division/written/game.shtml
http://www.bbc.co.uk/skillwise/numbers/wholenumbers/division/problemsolving/game.shtml

These games have been chosen for their content as the theme for the mathematical tutor to build has been defined to support the learning of divisions (*see chapter IV*).

⇨ Multipliotest
http://people.bath.ac.uk/sasg20/multiplio/

Whist Multipliotest had been originally created for French children at CM1-CM2 level, it has been chosen for his help system represented by an interface persona displaying emotions. Translated and adapted for year 3-4 children, it provides good quality assessment material on their perception of the representation and the usefulness of an emotive interface persona as a helper in tutoring systems.

The evaluation of the existing tutoring systems has been undertaken by the following evaluators:

⇨ Pascal Orzabal, our Participatory-Design child throughout the design process, evaluated all systems using a think aloud protocol. This method has been chosen as it is considered a highly efficient usability method for children aged 9-11, producing more results than post-task interviews or heuristics (*Baauw&Markopoulos, 2004*).

⇨ Patrick Girard[1], an HCI expert, also evaluated all systems to improve the effectiveness of the evaluation and reduce any possible bias or error. He chose a heuristics approach to evaluation based on Nielsen(*2001*) and Dix et al(*2004*) heuristics.

[1] Patrick Girard is professor for the department of computer science, director of the HCI research laboratory at the university of Poitiers (France).

⇨ The evaluation of Multipliotest will also be performed by the 2005-2006 year 6 class at Bathwick St Mary CE Primary School, Bath using an interviewing and questioning approach to get some insights on the use of a helper in tutoring systems and its need to represent emotions. Some smileyometers (*Read et al, 2002*) were used in the questionnaires (see section IX.2.4) to give a measure of expectation on the perception of the child of the activities and the emotional display of the interface persona.

See section IX.2.3 for a description of the evaluations and a summary of the findings from the existing system evaluation. The requirements elicited from this activity have been adapted and integrated with those ones highlighted by the other complementary elicitation activities in an initial set of requirements, which validation is described in the next section.

III.6 Requirements validation

One mean by which requirements validation can be undertaken is through a close collaboration with stakeholders. By negotiating on the content of the requirements with teachers, it is possible to iteratively refine the requirements for the prototype to include functionalities suitable for the classroom teaching.

This approach was adopted within this project as the requirements document was taken into Bathwick St Mary CE Primary School, Bath where input form Matt Cottrell was given to ensure that the set of requirements were suitable to support teaching and learning techniques currently employed within the classroom.

Matt Cottrell stated that the requirements were generally satisfactory and thought that the resulting system should be very beneficial to the classroom domain. He emphasized the need for his involvement throughout the design process to adjudge the mathematical consent and suitability of the tasks designed for children, as well as the interface design. Matt Cottrell also stated some recommendations for five of the requirements.

A full specification of the validated requirements for the tutoring system is detailed in the Requirements Appendix section IX.2.5. This provides a reference for the full set of requirements for the mathematics tutoring system, gathered from all requirements analysis activities, where the preceding modifications have been applied.

III.8 Conclusion

Through a detailed analysis of the needs and aspirations of the stakeholders of the project as well as the elicitation of the successful and unsuccessful features of current systems and past research projects, a stable set of requirements has been produced and validated. They form the basis for the design of the prototype solution which process is described in chapter IV.

IV. Design

IV.1 Introduction

As stated by Preece et al (*2002*), “Design is a practical and creative activity, the ultimate intent of which is to develop a product that helps its users achieve their goals.” The overall aim of the project is to build a software architecture to be deployed in schools as a main equipment basis for later research studies. The main objective of the design is therefore to develop a highly usable, interactive learning environment for children to practice effectively mathematics on a computer. From the point of view of the child, the user interface is the main interaction, as it provides a domain for them to interpret and execute system tasks. Consequently, the design stage involves making the interface intuitive and exciting so that children can easily relate to it.

The end system should be supportive of the tasks that the children and teachers perform and designed to be attractive for children. We therefore worked closely in the design process with a year 6 child and a year 6 teacher following a Participatory-Design approach to incorporate into the design their views and opinions about the user-interface and mathematical content.

In this section, we will first describe the Participatory-Design approach we undertook and the evolution of design produced from the Participatory-Design sessions. We then will detail the high-level decisions taken on the design stage. Prior to conclude, we will present some features of the detailed design.

IV.2 A Participatory-Design approach

IV.2.1 Introduction

The Participatory-Design sessions included the participation of two types of end users of the system: Pascal Orzabal, a year 6 child from Clifton College, Bristol and Matt Cottrell, a year 6 teacher from Bathwick St Mary CE Primary School, Bath. We will now present the evolution of the design with the different participatory-design sessions undertaken.

IV.2.2 First contact, 20th June

Pascal was chosen for the participatory sessions for his wide experience of using mathematical tutors on computers and his interest in playing games and creating artistic design on computers. The first session took place at his parents' home on June, 20th.

It consisted in an interview on Pascal's interests as for reading, playing on the computer or on console games, as well as his hobbies. This helped gaining a better understanding on the type of software he would like to be developed.

He then showed me the various kind of mathematical or tutoring software he uses at school and at home. An observation of his use of the software revealed excellent navigational skills and an interest in expressing the faults or originalities of the software user-interfaces and interaction styles proposed to the user.

We then reviewed the year 6 syllabus and the mathematical problems he might have encountered to define the mathematical content of the future software. Being an excellent student, Pascal faced only little difficulties in the learning of long divisions. We therefore decided to select the learning of divisions as for the mathematical area to practise with the computer, subject to approval from the Participatory-Design teacher.

At last, we performed a brainstorming session, expressing what he would like the computer game to represent and include. He quite instantly came with a clear idea of what he would like as a mathematical tutor: His interests in adventure and fantasy stories directed him to a fun and exciting "DividingQuest", where you could "change the world by succeeding mathematical challenges". Inspired by the Lord of the Ring™ storyline, we developed the idea of software with various mathematical challenges to succeed in different areas of the game, in order to finally reach the most challenging area and accomplish the quest. See section IX.3.2 for a description of the outputs produced by the participatory session.

IV.2.3 Existing System evaluation, 23rd June

The second child participatory-session aimed to elicited requirements from the evaluation of existing systems. A selection of tutoring systems was realised prior to the session from the list of software Pascal usually used in the BBCi website as well as a further software on the use of interface persona with emotive affects. During the evaluation, Pascal noticed the bad and good features of the existing software, adding naturally some comments on how they should be modified for the development of our DividingQuest (see Chapter III for a full description of the software and the evaluation performed). After the session, his comments were sorted and a list was drawn of the first "design solutions" to be exploited later in the design process.

IV.2.3 Defining the storyline, 26rd June

The third session was based on the first design of the software prototype. As the Participatory-Design was realised with only one user at a time, we decided on an adaptation of the PICTIVE and CARDS methods (*Muller, 1991, 1992; Tudor, Muller and Dayton, 1992*) to create a low-fidelity prototype. The prototype was chosen to be low-fidelity in terms of interaction and high-fidelity graphically, with the use of Microsoft PowerPoint™ (see sections IX.3.3 and IX.3.4, Design Appendixes). PowerPoint™ was also chosen as Pascal already had a good knowledge of its functionalities in creating animated cartoons.

A screen mock-up (a blank internet browser window) was used as a canvas on each PowerPoint slides for designing the system. The plastic coloured cards and icons of the PICTIVE method were transformed into square PowerPoint components with a coloured or pictured background to be moved around on the screen and on the various slides. Choices of pre-designed headers were also presented to Pascal, for him to place his favourite heading on the screen where appropriate. He chose the Apple Chancery heading to be placed on the center of the screen with the logo of the DividingQuest as a background.

The first activity consisted in the refinement of the storyline and the definition of the main screens of the game (see sections IV.3.4, IV.3.5 and IV.4.2). The focus of this exercise was to get an idea of the sort of colours, sounds, graphics and layout that he expected from the system. Pascal felt the need to make some comments on the various screens about the navigational structure or the content of the animations using the speech bubbles of PowerPoint™ as Post-It™, complying with the usual use of post-its in PICTIVE to signify objects that might move around on screens.

To follow, Pascal was asked to define the navigation buttons to be put on most of the interface screens as well as the menu icons (see section IV.4.2).

Given the interface components, Pascal designed the screens by distributing the icons appropriately across the interface, modifying the colour of each component in duplicating the slides to see different versions of the screens. He then refined his choice to one design solution only, with a good use of bright coloured interfaces.

The session ended with the definition of the different interface personas to be used within the DividingQuest as well as the emotions they should display. It was decided that the characters would differ in their species and sex. It was also decided that the colours of the interface persona clothes would change in the game according to the child's performance with reference to use of the traffic-light system (ie. green for success, orange for partially understood and red for a lack of success) in Pascal's class (see section IV.4.3).

IV.2.4 Defining the content, 3rd July

After approval of the subject of practising divisions and a definition of the main areas of difficulties of children in learning divisions by Matt Cottrell, the next session with Pascal aimed to define the mathematical activities of the areas of the DividingQuest as well as the help he would like to get on the different challenges.

A second activity was the definition of the number of rewards to get to be allowed to challenge the dragon (see section IV.4.1) as well as the content of the verbal praise given by the help system.

At last, Pascal expressed the will to access to the mean of results of other children in the child interface. He thought that the fact to visualize the class performance in an activity could motivate the children to do better on the tests and to practise more, adding also some motivation as for the challenge to succeed the better.

IV.2.5 Evaluation of the DividingQuest, 12th July

The first evaluation of the child interface prototype was realised by Pascal on July, 12th. The low-fidelity prototype was presented to him for his final approval on the design decisions made on every screen of the child-user interface according to the storyline defined.

He made some comments on the progress indicators of the mathematical quiz and finally decided to place them at the top of the frame instead of at the bottom for a better visibility.

IV.2.6 Children prototype validation and Teacher prototype creation, 15th July

The first teacher Participatory-Design session with Matt Cottrell took place on July, 13th. It consisted in the validation of the child user-interface prototype as well as the creation of the teacher user-interface prototype.

As an overall, Matt Cottrell found the child interface well designed and thought the children will be very interested in using the software. He validated the mathematical content and defined the content of the general methods to be given as a help system to children to match the class teaching practises (see section IX.3.4, Design Appendixes).

However, he advised two main design modifications concerning the passwords to access the site and the visualization of other children's results. Given the fact that the children are used to frequently forget their passwords while accessing to the computers, the school adopted the policy of one general password for the class. Matt Cottrell advised that the teacher chooses a general password for his/her class, the identification of a child being made through the use of a unique username. Taking into consideration that some children might have a lack of motivation in trying the activities if they realised the class average performance was hugely better than their own performance on the system, Matt Cottrell advised that the visualisation of the class results to be restricted to the teacher user-interface. He considered as really beneficial for his teaching to have access to the children results but did not want the learner model of children to be open to peers.

The last activity of the session was the definition of the teacher user-interface. It took place in the form of questioning about the functionalities the interface should include and the way he wanted the information to be presented. A low-fidelity prototype was then designed for approval in the next Participatory-Design session of July, 20th.

IV.2.7 Multipliotest experiment, 20th July

The second teacher Participatory-Design session was realised on July, 20th, after the evaluation of Multipliotest by the year 6 class of Bathwick St Mary CE Primary School, Bath (see section III.4). Matt Cottrell first commented the evaluation performed and the impact of the use of the software on children and then validated the teacher user-interface prototype.

IV.2.8 Requirements validation, 24th July

The last teacher Participatory-Design session took place on July, 24th with the validation of the requirements document (see section IX.2.5, Implementation Appendixes).

A list of design solutions to the final requirements document was drawn from the Participatory-Design session inputs and served as a basis for the implementation of the prototype solution (see section IX.3.6, Design Appendixes).

Several evaluations of the software were realised during the implementation process by Pascal in the summer from which some graphical changes were undertaken on the software. The final evaluation of the software took place on August, 26th 2006 (see section VI.5).

IV.3 High-level decisions

A number of high-level decisions have been made in the design process in order to define the scope of the design and implementation:

IV.3.1 High-fidelity prototype solution

As detailed in section IV.2.3, a low-fidelity prototype has been developed for the initial design of the system. Designed in Participatory-Design sessions, it served as a basis for the evolution of a high-fidelity prototype solution, encompassing the knowledge gained from the requirements elicitation activities (see Chapter III).

The prototype solution will be implemented using "vertical prototyping", providing complete functionality for only a subset of the available system functionality (a restricted number of mathematical challenges will be created).

IV.3.2 Mathematical content and context

Designed to include 5 areas of dividing challenges with 3 levels of difficulty in each area, the mathematical content of the computer-based learning tool will support mathematical content of the syllabus, ensuring consistency with concepts and methods taught in the classroom, as requested in functional requirement 9.

IV.3.3 Interface persona

The architecture of the software should enable investigations on the impacts of using emotive interface personas as helpers. The DividingQuest will therefore include 3 representations of the child user-interface, to be differentiated in their representation of helping system.

The first interface will present help to the user in the form of buttons. The second interface will use interface personas as helpers, giving verbal praise and rewards. The third interface will based the helping system on the second interface, with the interface personas expressing some emotions according to the performance of the child. In the second and third interfaces, the child will be asked to choose a helper in the 6 interface personas proposed.

The software will register their choice for further analysis on children identification with the characters as requested by functional requirement 22.

IV.3.4 System architecture

The system has a hierarchical architecture, with three different user-interfaces at its root: child, teacher and administrator.

The structure of the child user-interface can be summarized as follow:

⇨ The child connects to the DividingQuest with its username and password

⇨ If the child uses the software for the first time, he is presented with some options to choose.

⇨ After an introductory screen, the child accesses to the map screen and the menu bar summarizing the interface functionality.

⇨ He can then access to the mathematical quiz by selecting an area of the game.

In the teacher and administrator interfaces, the user is offered a choice of options according to the system functionalities.

Figure 4.1 represents a high-level model of the system for the child user-interface:

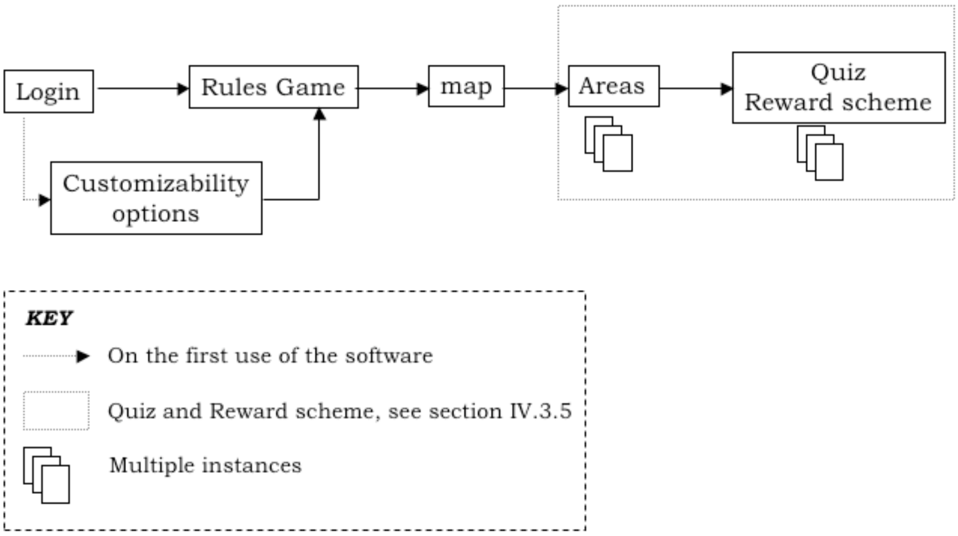

Figure 4.1: Child user-interface system architecture

IV.3.5 Quiz architecture

Each quiz of the game follows the same architecture illustrated in Figure 4.2:

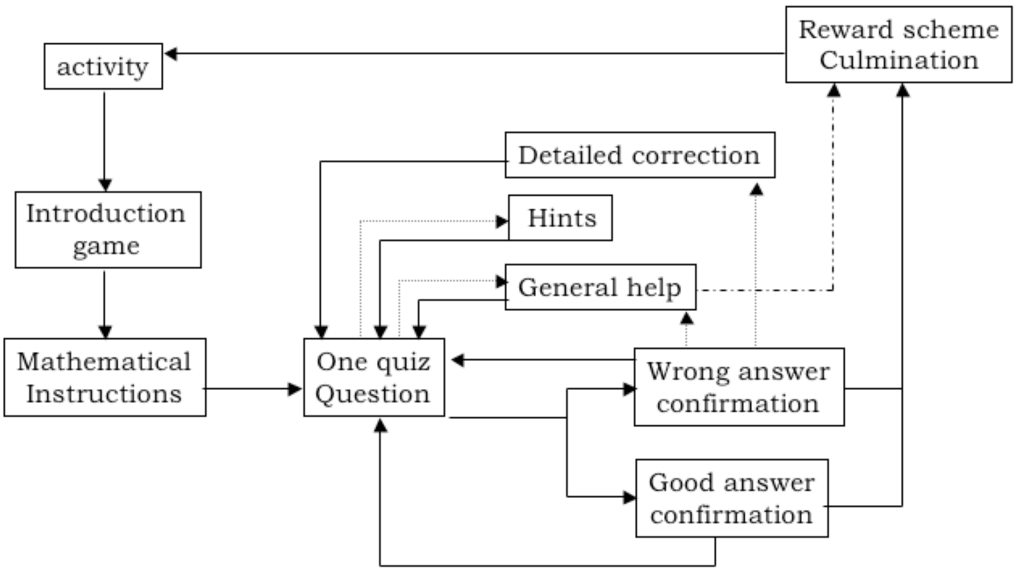

Figure 4.2: Quiz architecture

The quiz structure can be summarized as follow:

- ⇨ The user selects one level of quiz in the three levels of the area selected.
- ⇨ The goals as well as the rules of the quiz are introduced.
- ⇨ The user is reminded of the key mathematical concepts needed for the quiz completion.
- ⇨ The user is presented with a question to solve. They answer this question by using point-and-click and form-fill interaction.
- ⇨ If the user spends too much time answering the question, the help system advises them to use the general method helping screen or some hints on how to answer the question.
    - o After accessing to the general help or the hints, the user is taken back to the question to answer.
- ⇨ After answering the question, the answer is checked against the actual answer and one of the following screens is presented:
    - o An animation telling the user they have selected the right answer, which links to the reward scheme
        - ▪ By selecting the next button, the user is taken to the next question.
    - o An animation telling the user they have selected the wrong answer, and giving them the correct answer. The user is advised to look again at the general help method or at the detailed correction of the question before to go to the next question.
        - ▪ After accessing to the detailed correction or the general method, the user is directly taken to the next question.
        - ▪ By selecting the next button, the user is taken to the next question.
- ⇨ After the user completes the last question of the quiz, an animation linked to the reward scheme is displayed with the possibility to access the general help one more time.
    - o After accessing the general help method, the user is taken back to the end of quiz screen.
- ⇨ The user accesses to the level selection screen with an update on his performance according to the result of the quiz.

The user will be able to stop the quiz at any time by selecting a menu icon outside from the quiz area.

IV.4 Detailed design

The detailed design section covers some of the specific features of the design of the DividingQuest, including the storyline, the representation of the quiz, the progress indicator and the design of the choice of helpers screen.

IV.4.1 DividingQuest storyline

The storyline was first defined as a warrior's quest, the mathematical performance enabling the warrior to gain some "life points" to be able to fight the dragon.

It was then decided to change the reward scheme to add more challenge and interest in the game (section IV.1.3). Some of the mathematical quiz being more difficult than others, it was decided a separation of the mathematical activities into 5 areas, containing 3 levels of challenge each. Winning a level 1 quiz gives the child the access to one particular piece of its armour made of bronze. Winning a level 2 quiz transforms this piece of armour into silver whereas winning a level 3 quiz transforms the silver piece into gold.

To access the final part of the game (the Challenging the dragon Area), the warrior should have some knowledge in each of the mathematical area (5 pieces of armour), being more familiar with at least 2 areas (at least 2 silver pieces) and a specialist in at least one of the areas (at least 1 gold piece).

IV.4.2 Quiz scenery

The main game-play takes place in the form of mathematical quiz. In order that the interface did not distract the users from the task, the quiz scenery has been designed as simple as possible, following the structure illustrated in Figure 4.3:

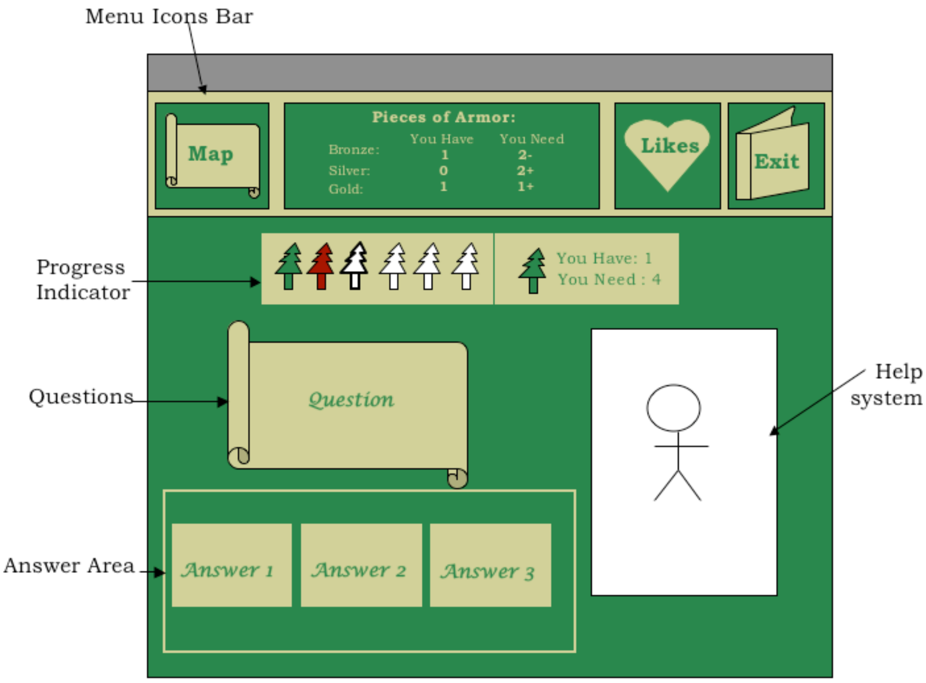

Figure 4.3: low fidelity mock-up of the quiz

Excluding the help system area, the main areas of interaction are placed in the centre of the screen to give the children focus within the scene. The progress indicator bar is placed at the top of the screen, the menu frame being placed at the top of the quiz scenery, with the reward board and the system functionalities.

IV.4.3 Progress indicator

As defined in section IV.2.3, the progress indicator of every areas of the quest is based on the traffic lights system. The graphics however differs from one area to another to match the environment of the area they belong to and help the children remember in which area of the game they are. Figure 4.4 illustrates the progress indicator for the *Bewitched Woods Area*:

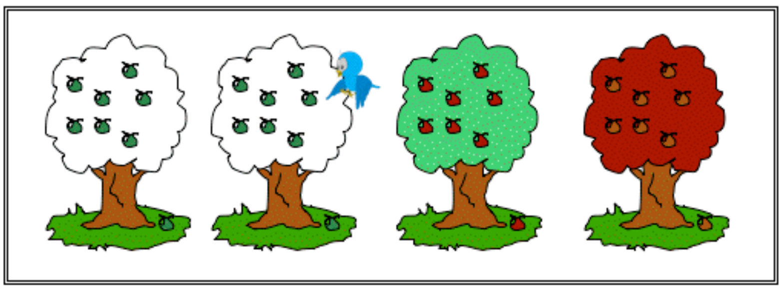

Figure 4.4 Progress Indicator Icons

A white icon with an animal has been added to represent the current question being answered and differentiate it with the questions still to answer.

The traffic-lights system (see section IV.2.3) has also been used in the representation of the interface personas to magnify the verbal praise given after one question or one quiz: the clothes are progressively transformed from bottom to top to green if the child answered correctly or succeeded the quiz. On the contrary, in the case of an incorrect answer or an unsuccessful test, the clothes turn progressively to red from the top to the bottom.

IV.4.4 Choosing a helper

The screen asking children to choose a helper was first designed by our Participatory-Design child as follow: the screen would present the helpers in three columns, with their graphical representation and their name underneath the image. The helper would be selected using a radio button. When willing to access some more details on a particular helper, the child would click on the image and a popup window would appear with a full description of the helper's characteristics.

As the use of radio buttons is firmly discouraged in the HCI literature (following Fitt's law), this part of the low-fidelity prototype was redesign with the child as follow: every helper will be presented in the screen with the character's names underneath their image. By clicking on the image of a helper, the user will access another screen with the description of the helper and the possibility to choose this helper or to go back to the previous screen and access to the description of another helper.

IV.5 Conclusion

Using a child and teacher Participatory-Design approach for the design stage, an iterative and detailed design has been realised in consideration of the user's needs and desires, enabling us to formulate a final specification of the design solutions to the system requirements (see section IX.3.6, Design Appendixes), which have been directly applied when implementing the high-fidelity prototype.

The child Participatory-Design sessions enabled the definition of an interesting storyline for the game, all the interactions within the software being designed with Pascal’s help to enhance the appropriation of the software by year 6 children.

The teacher Participatory-Design sessions helped the inclusion of a mathematical content adapted to the classroom teaching and a customization of the children and teacher models to teacher’s expectations in order to increase the pedagogical value of the system.

Chapter V denotes the essential stages of the implementation of the system which architecture aims to serve as an equipment basis for further research analysis.

V. Implementation

V.1 Introduction

In this chapter, we detail the implementation process that leads to the development of the high-fidelity prototypes version of the DividingQuest, to be used in the future as the basis equipment of an experimental study. The organisation of the chapter follows the implementation process. We will first expose the rationale for choosing the implementation language. We then will detail the software architecture and the decisions made concerning its user model and database components. Prior to conclude, we will describe some particularities of the user interfaces.

V.2 Implementation language

In both the requirements and design chapters we discussed and specified the need to produce a highly interactive, usable and highly graphical high-fidelity prototype with audible output and direct manipulation interactivity. Taking this specification in consideration, there are only a limited number of software packages and programming languages that can be used to develop the prototype, including:

- ⇨A web-based software (PHP, HTML, Coldfusion, Java Applet) with a database back-end.
- ⇨Java
- ⇨Macromedia Flash with Action Script programming language
- ⇨Microsoft Visual C++.Net

Both Java and C++ would require an appropriate compiler and virtual machine to be installed before the system could run which goes against the functional requirement 1 which states that the software should run as a standalone program without the need to install any additional software. To fulfil functional requirements 22, the prototype will need to store some data about the user, it was therefore chosen to implement the prototype as a web-based program developed in PHP and HTML with a SQL database back-end.

It was also decided that Macromedia Flash 8 would be used to create the animations that will be included in HTML components.

The development of the prototype has been realised using Macromedia Dreamweaver and MAMP (Apache server) for the website, phpMyAdmin for the SQL database and has been tested on Safari, Internet Explorer and Mozilla. (see section IX.4.10, Implementation Appendixes for the publication of the software).

V.3 System architecture

V.3.1 Website Architecture

As highlighted in the design chapter, the system is composed of three different user interfaces (child, teacher, administrator) that include various functionalities and zones for interaction. For the system to be easily reused by developers, the website files follow a hierarchical tree corresponding to the diverse areas of the software. All graphics, sounds and animations used by the system are classified according to the same structure in an *images* folder at the website's file root. As the number of files created for the software goes over 100 files, we will not present a detailed description of the architecture. See section IX.4.2 in the Implementation Appendix for the network diagram illustrating the navigational structure of the system.

V.3.2 Database and user-modelling

To facilitate the use of the software as a basis of an experimental study, there has been a need to store data about the user of the system, keeping a trace of the user's actions and achievements. For this purpose we carefully designed a SQL database in the third normal form to register the interaction of children and teachers with the DividingQuest. See section IX.4.3 in the Implementation Appendix for an entity-relational diagram of the database.

The child user model is initialised in the teacher's interface. Each child chooses its helper during its first use of the software. Furthermore, the child can modify it's likes when convenience. The child can then access some information about its user model such as the number of rewards it won or the number of questions to succeed to get a particular reward, but cannot modify its user model. The teacher can however act on the child's user model in her/his interface by setting the number of questions a particular child needs to succeed to get each reward. He can also access the results of every child in his class and modify his own personal data. The child user model is therefore only open to the teacher and the child (with some restrictions) as stated in functional requirements 10 and 21.

Given the fact that the software contains personal information about the child and the teacher's user model and achievements, it was important to secure the access of the information of the website to the authorized people. (see section IX.4.4, Implementation Appendixes for a description of the two security system implemented).

V.4 DividingQuest Detailed Implemenation

In order to promote the design principles of familiarity and consistency (*Nielsen, 2001; Dix et al, 2004*), it has been necessary to begin the software development by defining a template for the system components (see section IX.4.5, Implementation Appendixes). In this section, we present some particularities of the system implementation.

V.4.2 Warrior Interface

The warrior interface represents the main interface of the system. It includes three different child user interfaces corresponding to the three representations of help system (with buttons, with interface persona and with emotive interface persona). It proposes 15 mathematical activities about learning divisions, including 5 main areas with 3 levels of difficulty. All user-system interaction in the system differing only in content in the various activities but not in structure or in the data registered, we developed the activities incrementally.

<u>Questions</u>

Each time the user tries a level of an activity, a random selection of the questions to answer is made. For one specific question, the possible answers try to match the method to be applied to solve the question. The algorithm that produces random possible answers for the first activity produces answers that can be estimated using the technique presented in the general help method. Once one pattern of solution has been chosen, it randomly assigns the positions of the possible answers so that if the next question follows the same first pattern, the positions of the numbers will differ. The user will therefore not be able to guess the answer from one question to another (see Sample 3, Implementation Appendixes, section IX.4.7).

When choosing a possible answer, the children click on its button and have an immediate feedback on the result of the question. Figure 5.2 shows a question to be answered with three possible answers:

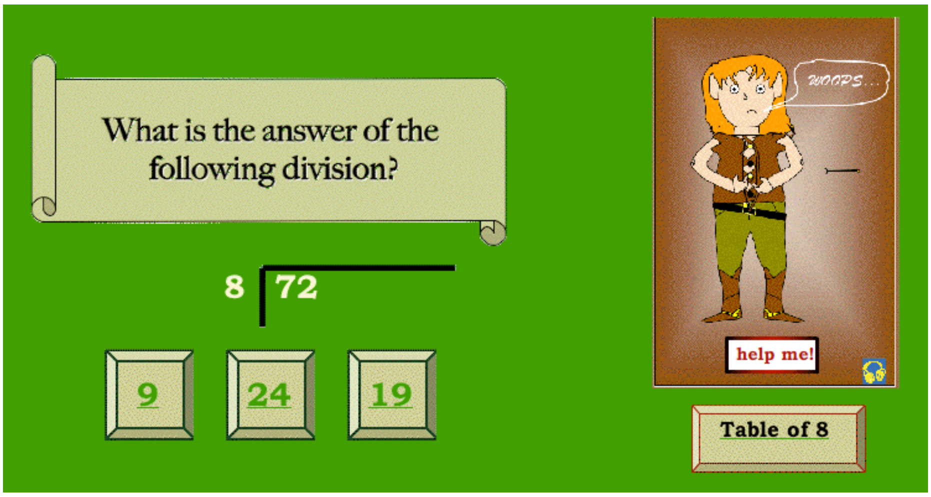

Figure 5.2: Question with 3 possible answers

To fulfil the functional requirement 22, the software should register the child performance in the quest according to the guidance they have been given for future analysis in an experimental study. See section IX.4.9, Implementation Appendixes, for a description of the mechanisms used to register the child's response to the advices given by the help system.

Progress indicator

As described in section IV.5.5, the progress indicator concerning every areas of the quest keep on the traffic lights metaphor but differs graphically for the user to remember in which area of the game they are (see Figure 5.4). See section IX.4.8, Implementation Appendixes, for a description of the session variables used for the progress indicator.

Loneliness Desert Area:

Stormy Sea Area:

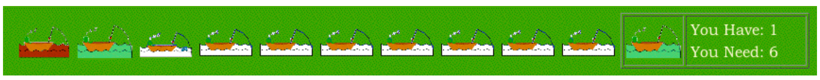

Figure 5.4: Comparison of Progress Indicator

Help

Three types of mathematical help are given to the user: a general method to answer the question, some hints (such as the access to division tables) and the access to a detailed correction of an incorrectly answered question. To fulfil the functional requirement 22, the child interaction with the system is registered for future analysis in an experimental study. When a child accesses some help, we look at the effect of using help on the result of the next question. The use of help and the result of the next question are then registered in the database.

Child User-Model Modification

⇨ When the user first uses the software, he is asked to select a helper in the case of representation of help system by an (emotive) interface persona. The software registers the choice of helper in the database. In the future, when the user will connect themselves to the database, the system will get the identification of the helper chosen and keep it in a session variable along with whether it displays emotions for the help representation throughout the child's quest. Once the child chooses a helper, there is no possibility to change for another helper.

⇨ The child can interact with his user model through the choice of things it likes the most. As detailed in section IV.4.4, it has been decided that the questions on problem solving would be exposed to the child using the things it likes the most instead of a random choice of themed pictures and concepts. For this reason, the user if asked to choose a number of things they like in different categories at their first use of the software. Taking into considerations of Fitt's law and screen overload, it has been decided to present the user with a succession of screens representing options for one activity only as the repetition of screens will appear only once for all (at their first use of the software). Once they chose their likes, they can click on the 'I have chosen' button and access to the next category or the main frame if all categories have been completed. Information is registered in the database for future use in the problem solving activities. Children can change their selection by clicking on the 'Likes' button on the top menu frame at any time. To facilitate the user accessing to one particular activity for a modification and to limit the number of screens to go through, it was decided that this second likes interface presents all information on the same screen, in two category columns. The software gets the likes registered for the child and displays the categories with the choices already made by the user.

V.4.3 Teacher Interface

The teacher interface contains a few functionalities concerning the teacher's personal information. However, its main functionalities concern the child OLM with the ability to create a class, adding children to a class, access information about the children of his/her class and adapt the children learner-model.

Creation of a child

When the teacher creates a new child belonging to an existing class, he has to enter the following information about the child: name, forename, sex and grade (the average child level, set between low, medium and high). At the creation of the child, the system will create a unique username, combination of the number of children already existing in the database with the same forename (see Sample 4, Implementation Appendixes, section IX.4.7). The system assigns a condition for the help representation given to the child for the number of children with the same grade to be as equal as possible in each of the three conditions (see Sample 5, Implementation Appendixes, section IX.4.7). The child is then linked to the class it belongs to in the database.

Child OLM considerations

As previously stated, the child learner model is open to the teacher only. S/he has the opportunity to visualize the performance of every child of his/her class and can also act on a particular child user-model.

One option given to the teacher is to visualize the number of children that have worked on the various activities and the number of children that have won rewards. The software sums up the rewards of each child of the class and displays the result according to the activity and the type of reward.

The same technique is used to present the teacher with all the performance for a particular child. Associated with the rewards won and the numbers of attempts to won the reward is the number of questions the child has to answer correctly to succeed the quiz with the number of questions that comprise the quiz. These two last numbers can be modified by the teacher as requested in the design process (section IV.5.7). The modifications are then inserted into the database and a link is made between the child and the teacher to register the modifications.

V.5 Conclusion

The implementation process led to the development of a high-fidelity prototype functional enough to be evaluated and deployed within primary schools after a proper system testing (see section IX.4.11, Implementation Appendixes). For the purpose of testing, evaluation and future user of the software, the website is accessible at the following address: http://people.bath.ac.uk/sasg20/DividingQuest/, children and teachers accessing their user interface once a username and a password have been attributed to them.

In the chapter VI, we will detail the usability evaluations performed on the prototype in order to highlight areas in which the tutor meets the required functionality for the users, and areas that require further development.

VI. Evaluation

VI.1 Introduction

As stated by Preece et al (2002), "Users want systems that are easy to learn and to use as well as effective, efficient, safe and satisfying". Evaluating the prototype solution is therefore an essential stage of the project, as it will assess the extent to which the solution meets the key objectives and system requirements defined in the development stage.

In this section, we will first present the evaluation undertaken with respect to the requirements document defined in chapter III and assessing whether the tutor fulfils all of the desired functionalities. We then will expose the usability evaluations performed, assessing whether the implemented solution utilises the design principles and heuristics discussed in the literature review as well as meeting the requirements. To follow, we will detail the redesign process undertaken from the conclusions of the former evaluations as well as the suggestions for future improvements on the design. We will conclude on the results of the final evaluation undertaken by our participatory design child on the prototype solution.

VI.2 Evaluation against the requirements

The evaluation of the requirements document outlined a categorization of the evaluated requirements under three broad headings: the fulfilled requirements, the requirements whose functionality was partly implemented and require further development, and finally the requirements that were not implemented during the project's development and are considered as future enhancements to the system.

The results of the evaluation are presented following the categorization requirements used in the requirements document.

Functional requirements:

All functional requirements were fully implemented during the prototype with the exception of requirements 3 and 8 that would need further development.

Whilst the user has the possibility to skip lengthy animated sequences as requested in the requirement 3, the introductory screen explaining the storyline and the rules of the game at the beginning of the software does not provide the ability to stop the animation and begin the instructions again. As the sequence is quite long, a design improvement might be to add a "repeat from the beginning" button.

The inclusion of navigational instructions and mathematical explanations within the tutor enables the prototype to fulfil requirement 5,which states that the system should allow the child to work alone without the need for teacher interaction. The provision of this support indeed reduces the likelihood that pupils will ask for assistance in answering questions.

The design of the tutor included a different content for each level of the 15 activities presented as requested in requirement 8. However, only 5 of the activities have been implemented in the prototype solution, the other activities to be developed in future versions of the software.

As stated in requirements 17-20, the tutor must provide audible instructions as the primary mode of communication with the user as well as a highly graphical interface. Implementation of the animations of the system using Macromedia Flash 8 has enabled the development of a highly interactive and graphically intensive tutor, which therefore enables these requirements to be fulfilled.

Non-Functional requirements:

All non-functional requirements were fully implemented within the prototype solution with the exception of requirement 1 and 7 that need further development. Requirement 14 was designed but not implemented as it is part of the activities to be developed in a future version of the software.

Requirement 1 states that the content of the tutor should progressively become harder, incrementing in small steps. Whilst the content of the tutor fulfils this requirement in terms of the activities proposed, all questions within the activities are created randomly.

Part of the requirement 7 states that the software should distinguish between a wrong answer and a lack of answer given. Whilst this functionality has been designed, this concerns some areas of the game that have not been implemented yet, every activity developed proposing a multi-choice of answers to the child with no possibility to skip one question as requested in requirement 11.

Usability and user-interface requirements:

All of the usability and user-interface requirements were fully implemented within the prototype solution.

As stated in requirement 3, all screens design and usability guidelines were adhered to throughout the design of the system, using a WIMP point and click interaction style as stated in requirement 1.

User requirements:

All of the user requirements were fully implemented within the prototype solution.

As stated with requirements 2 and 3, the tutor must maintain the attention of its user by supporting the user's motivation and enjoyment as well as following a storyline. The implementation of the reward scheme following the quest storyline along with the use of highly graphical animations is of particular relevance to these requirements, and a good source of motivation.

The evaluation has established that all requirements have been met by the prototype, with the exception of those considered for future software enhancements (the development of activities of level 2 and 3).

VI.3 Usability evaluation

As discussed within the literature review and in the requirements analysis, it is essential that the system is usable and does not produce any unexpected behaviour to ensure the maximum learning outcomes are obtained by the children.

In this section we present the results of the two analytical usability evaluations undertaken on the system to identify any particular good aspects of the system that clearly fulfilled the system requirements, or any particularly bad aspects that would require redesign for the future.

The first evaluation has been realised using usability heuristics by *Patrick Girard*, the HCI expert who evaluated the existing solutions for the requirement analysis.

The second evaluation was performed using the co-discovery evaluation method by two French young person: *Constance Allo*, 13 years old and *Claire Girard*, 14 years old. Whilst they were out of the age range of the software users, Constance and Claire were chosen as evaluators for their expertise in using mathematical tutoring systems (Adibou™, Adi™ series of pedagogical softwares, Dr Brain…) and their creativity skills providing valuable evaluation skills on the graphical and interactive components of the software.

Prior to the evaluations, the participants were given a small verbal introduction to the system. This enabled them to become aware of the main features of the system, the main aim of the tutor and the reward scheme provided.

VI.3.1 HCI expert evaluation

Heuristics evaluation coupled with a think aloud method were performed by Patrick Girard, an HCI expert on 10th August 2006. See section IX.5.2 in the Evaluation Appendixes for a full description of the evaluation results.

The evaluation revealed a good interface design with a good navigational structure and no critical problem were encountered. In the following, we present the main good and bad aspects of the tutoring system:

⇨ Whilst the possibility to exit the system at any time is provided, no exit confirmation screen is proposed to the user, which goes against usability and user-interface requirement 6.

⇨ On the first introductory screen explaining the rules of the game, the button 'next' is not visible, children have to scroll down to be able to access it.

⇨ Some of the buttons in the activities screens are not active, without explaining to the child the reason why they cannot access the upper levels.

⇨ Whilst containing a sound layer, some of the animations do not present any graphical sign of audible presence whereas others do which goes against interface consistency.

⇨ When modifying the likes, after clicking on the "I have chosen" button, no feedback is given to the user to specify that the modifications have been done.

⇨ The user feels in total control of the system and the user of graphics and animations is used appropriately to motivate the child without distracting them from the task at hand.

VI.3.2 Children evaluation

In order to evaluate the usability of the tutor, consideration was given to the evaluation method that would provide the basis for effective evaluation.

We used a co-discovery evaluation method coupled with a post task heuristics evaluation, method considered as appropriate for encouraging children to verbalise what they do and think during interaction (*Markopoulos &Bekker, 2003*), As stated by Kemp and van Gelderen (*1996*), the co-discovery method consists of two users working collaboratively on tasks set by the usability specialist, which is assumed to produce better results as users will provide comments more naturally when they explore the interface together. See sections IX.5.3 and IX.5.4 in the Evaluation appendixes for the task sheet defined for the co-discovery evaluation and a full description of the evaluation results. The co-discovery evaluation was performed by Claire and Constance on 15th August 2006. Whilst the evaluators were not English, their knowledge of the English language was more than sufficient for them to appreciate the content of the software.

⇨ Overall, the system was considered to include a good storyline, with fun and motivational animations and helping characters to help the children in their quest. The aim of the warrior to conquer his world again by helping the people from the various areas gaining their knowledge back was particularly appreciated.

⇨ The system included a good visibility status scheme with the use of the map, the reward board and the activity progress indicator.

⇨ Users can make good use of the map button in the menu frame to exit a quiz at their convenience and thus not to be locked up into an area of the game.

⇨ The choice of the helpers and the possibility to select what they liked the most was also appreciated. However, the evaluators wondered what would be the future use of the likes, as they did not have any effect on the software during the evaluation.

⇨ The animations were found to be motivational without distracting the user from the task at hand. The general help method given was found to display information at a good pace for the user to understand the method with a good succession of learning steps.

The usability evaluations have shown that the system is globally usable and that no issues arose which prevented the children from being able to successfully complete the tasks on the system. As both evaluators were not familiar with the English mathematical teaching practices for year 6 children, no evaluation of the mathematical teaching methods within the tutor have been made. However, the mathematical content was designed with the help of Matt Cottrell and the implementation complied with the design of the mathematical interfaces. The next section present the design solutions identified for all of the issues that arose, in order to improve the usability of the system within a redesign.

VI.4 Redesign

In this section, we present the redesign of some features of the software advised in the former evaluations and the design considerations for the future version of the software:

A confirmation exit screen has been added to the system, with the possibility to cancel the exit, to exit the system or to access to an evaluation questionnaire.

The interface of the introductory screens has been modified to display the button next at the right of the animation and the back button at the left side of the animation.

When a child clicks on a button from an area of the game he cannot access because of a lack of rewards or performance in the previous levels, a javascript popup appears with a description of what the child needs to accomplish to access this area of the DividingQuest.

Each animation containing sound output presents a graphical representation of headphones on the right bottom corner of the animation.

After clicking on the ‘I Have chosen’ button in the likes screen, the user goes back to the map, the information being updated into the database.

As future design considerations, we would like to develop the problem solving activity taking into account the choices made by the user, as well as adding a button to begin again the instructions for the introductory screens, and to develop all activities of the system, taking also into account the performance of the child completing an activity in the display of the activity’s questions.

VI.5 Final Participatory-Design Child evaluation

A final exploration of the whole system, using the think-aloud technique, was undertaken by Pascal Orzabal, the year 6 participatory design child who helped us throughout the design process, on August 26th 2006.

We present here the feedback gained from this evaluation, in addition to possible redesign suggestions:

It was suggested that the help system could appear on the map screen to guide the user in their choice of actions, and decide which area they should try to get to the Challenging the Dragon area the ‘fastest’.

It was suggested that the hints given were removed if the child succeeded too well in the activity, to add some challenge to the game.

The content of the activities and the help provided were found to be of good value for the children learning and the implementation of the storyline within the activities matched perfectly the one defined with the child in the design process.

VI.6 Conclusion

From the evaluations performed, it was apparent that the users showed a clear understanding of how to use the tutor, and did not require any prompting or guidance on how to navigate the system and complete the tasks. It also demonstrated that the requirements of the system have been met in a usable way, and raised some issues and associated design solution to be implemented in future versions of the system.

Co-discovery and think aloud evaluation methods have proved to be highly effective methods of evaluation and have enabled the elicitation of both the positive and negative aspects of the design and usability of the system.

Using the stakeholders of the project for the final evaluation also had positive implications as to the future design of the system, with conclusions made from both evaluations being effectively used to analyse the systems limitations and highlighting its possible improvements.

The next chapter concludes the project, taking into account the extent to which the aims and objectives of the project have been achieved, the project's outputs in terms of software architecture and considering the use of a participatory design approach for the design process. It finishes with the presentation of a future research program and an exemplar study to investigate whether there is a difference in learning achievement and motivation between a help system represented by buttons, an interface persona, or an interface persona expressing emotions.

VII. Conclusion

VII.1 Introduction

In chapter 1, we stated the importance of investigating the impacts the use of new technological devices have on children in order to better support them in everything they do. We also discussed the need for some learning practises enabling the most able children to be stretched academically, whist providing extra support as the basis for those who require it. Given the prohibitive cost of a one-to-one tuition within classes, it has been questioned whether the use of a mathematical computerised learning tool would help children learn better. Those considerations also highlighted the scope for researching how the inclusion of some kind of affect into the design of learning tools might help children have a better enjoyable and motivational learning experience.

The aim of the project was to provide software in the form of a Mathematical Teaching Help for year 6 children to be tested in an empirical study about the impacts of using interface personas with strong emotive representations as helpers in Open-Learner Modelling tutors on the appropriation and understanding of the user-model component by children.

Chapter II provided a largely theoretical grounding in the main issues relevant to the problem, as well as the justification for using a computer-based tutor for educational purposes. The literature review included an investigation into relevant HCI and CCI issues, which has formed an important underpinning to the project in order to make the system highly usable, easy to learn and relevant for year 6 children. Research in 'funology' and designing interactive systems for and with children highlighted the need for the stakeholders of the project to be included in the design process for the software to match its user's needs and aspirations, and to offer a pleasurable learning experience.

The requirement analysis described in chapter III therefore deeply involved the stakeholders of the project through naturalistic observations and questioning about teaching and learning practises. This gave us a better understanding of the problem domain and thus identified the issues grounded in both theory and practise. The existing tutoring systems evaluation enabled us to examine the ways in which these issues are currently supported with a view to developing a prototype solution that improved upon many aspects of their design.

The design process (chapter IV) was realised using a Participatory-Design approach using the inputs from both year 6 children and teachers. The Participatory-Design sessions incorporated elements of several existing Participatory Design techniques that were adapted and combined to produce a method corresponding to the available resources. This approach to the design process produced the main design ideas for the development of the high-fidelity prototype (chapter V), with considerations to the teacher and children needs and expectations. It furthermore helped defining a mathematical content adapted to the class teaching practises, more likely to produce an enjoyable and valuable learning experience during the software deployment in classes.

From analysis of the evaluation chapter (chapter VI), the prototype solution developed is clearly highly usable and interactive and ready to be deployed in schools for research studies which fulfils the main project aim. The children participating in the usability evaluations found the system fun and exciting to use, which demonstrated a good use of the Participatory-Design sessions.

In this section, we will first conclude on the achievements of the project in terms of the software architecture and the use of Participatory-Design in the design process. The report will end with our future research program, including an exemplar study to investigate whether there is a difference in learning achievement and motivation between a help system represented by buttons, an interface persona, or an interface persona expressing emotions.

VII.2 Outputs of the project

VII.2.1 The DividingQuest

The system prototype was implemented successfully, with the ability to register some data about the user interaction with the system.

The architecture presents researchers with three possible user-interfaces for children, a management system by teachers and a visualisation and customization interface for researchers on the variables to register and the user-interfaces to attribute to children.

Some design improvements and complements of implementation are presented in our future research program, in section VII.3.2.

VII.2.2 Assessment of the utility of participatory design

The results of this project would suggest that the key towards successful Participatory-Design is the iterative nature of the sessions including both children and teachers in the design process.

The organisation of the child Participatory-Design sessions had an iterative nature, for the granularity of the design being refined at each stage of the design. This enabled the iterative sessions to have a different focus each time, providing more motivation to the child as each session produced a result closer to the final product. Some suggestions however were contradictory to several design solutions generated from studying the HCI literature, as well as violating HCI practises. In these cases, the child's inexperience within the field of HCI was noted and existing professional theory was preferred.

The inclusion of teacher Participatory-Design sessions within the evolution of the design realised from the child's inputs were revealed particularly salient, some compromises being needed between design ideas from the children on the mathematical content or their access to peers results and the teaching strategy of the teacher and its pedagogical groundings concerning the openness of the child user model. It was therefore possible to modify the design from the teacher's input, explaining to the child the reasons of the modifications and enabling another iterative refinement of the design from the teacher's inputs.

In terms of activity content, the order of techniques used was found to produce good results, as the child was able to see where each activity was originated from and what remained to be done to achieve the design. The use of PowerPoint™ to create the prototype enabled the child to express the exact representation of the concepts and images he wanted to be included in the software, whereas the use of oral description might have produced a final design different from the child's expectations. It also enabled the child to make affordances to actual applications he had experience using and to have a better idea of the final result of the prototype solution according to his expectations.

Due to the experience of using computers of our Participatory-Design child, little adaptation were done on conventional techniques such as PICTIVE and CARD, emphasizing the user of Post-It™ to describe the interaction in the software. We however excluded some potentially useful elements such as the use of video recording due to the project timescales.

Our Participatory-Design analysis includes some limitations as only one child and one teacher have participated to the sessions. For future Participatory-Design, it would be interesting to investigate the results of the method used in this project with a group of children, possibly realising the Participatory-Design with a class for the initial design and a group of children outside from the class environment for the refinement of the design, as the later activity is more time consuming and could not be realised in the class due to time commitments.

VII.3 Further Research Program

VII.3.1 Design Study

The exemplar design study we will describe in this section will enable the investigation as to whether there is a difference in learning achievement and motivation between a help system represented by buttons, an interface persona, or an interface persona expressing emotions.

It includes a between-subjects longitudinal comparative study, which includes pre and post session learning tests. A second factor has been added to the design as there will be repeated-measures between the learning sessions. The experiment will evaluate the learning achievements and the motivation of children using the software against a lecture realised by a teacher over a period of time.

The experiment aims to answer the following research questions:

- ⇨ What are the benefits on learning and motivation of integrating the system into the classroom over traditional teaching methods?
- ⇨ Do children have a better experience and more motivation if they are able to choose an interface persona with which they associate?
- ⇨ What characters are chosen most often and what are the reasons for this choice?
- ⇨ Do children perceive and appreciate the emotional aspects of the characters?
- ⇨ What are the effects of chosen personas also displaying emotions on learning and motivation?

In order to assess the learning potential of the software the experiment will include one control condition and three levels of experimental conditions, the control group will consist of children being taught in the classroom by the teacher. Children from the three experimental condition groups will use the software respectively with a button, interface persona or emotive interface persona as representation of the help system. The children will be assigned to one group in order to have children with equal mathematical abilities in each group.

The experiment will include measures on the task completion performance, the performance using help and help guidance performance.

A pre and post mathematical test will be performed by both groups prior and after every session. The measures of the experiment will also be automatically registered after the sessions on a separate file.

An analysis to assess if there is a learning improvement of the use of each condition over the children being taught in the classroom will be realised after each session and at the end of the experiment. The former analysis will then be checked against to evaluate the effects of the different conditions on the learning of children. The analysis will also take into account the improvement on learning from the children on areas of the game they did not access during the learning session.

The same analysis will be undertaken for the groups making use of the software concerning the task completion performance, the performance using help and help guidance performance after each learning session and at the end of the experiment.

A questionnaire will be given to the subjects having used the software at the end of the experiment (see section IX.6.1, Conclusion Appendixes). The qualitative analysis performed from the questionnaires will assess the following research questions:

- ⇨ whether an (emotive) interface persona is perceived as more motivating by children
- ⇨ whether the condition with emotions has been perceived and appreciated by the children
- ⇨ the reasons why the children the children chose this particular character
- ⇨ whether the children feel more confident with the mathematical concept after using the software

Some other research questions can be raised from this exemplar study and the possible use of the software, to be described in the next section.

VII.4.2 Future Work

Some further research studies could be realised using the software architecture to answer the following research question:

⇨Does the use of emotive interface persona enable a better identification of the child with the characters?

A longitudinal experiment could be realised, giving the choice to children with (emotive) interface personas as helpers to change their helper in the middle of the game, analysing their performance using each helper and their willingness to change helper or to stay with the helper they identify the most with.

⇨Are teacher willing to give advice to the children through the use of the software and are children more likely to follow advices in the emotive condition?

A longitudinal experiment could be undertaken, displaying the helper on the main map frame and giving advice to the child. The advice could be given from the teacher's input in his own interface or a generation of advice according to the child's performance could be realised by the software.

⇨Do certain activities of the software contribute to a better children understanding of mathematical concepts than others?

A longitudinal experiment could be realised with the activities as the experimental conditions in the three user-interface representations.

⇨How children from different cognitive development respond to the help given?

Some cross-sectional studies could be realised diagnosing the childrens' cognitive development with the use of Arroyo's computer-based test (*Arroyo, 1999*), analysing the children responses to the abstractness/concreteness of hints, general help and detailed correction.

For the purpose of the project, not all the mathematical activities were implemented. A further development therefore could be a full implementation of the mathematical activities, as well as modifying the instruction model of the tutor to increase the difficulty of the problems it gives depending on the student's progress. The children could then have the opportunity to interact with their learner model and ask for easier or more difficult questions in the test. The results and learning benefits of this openness of the child learner model could be then analysed in further research studies.

Another software development would be to open some information on a class performance on the various activities to other schools registered with the system, for a class challenge to be organised. The children could therefore see as a challenge to succeed better in the quest than other schools, increasing their motivation in the learning session.

VIII. Bibliography

VIII.1 Books

- Acuff, D. S., & R.H. Reiher (1997) *Kids' Stuff: Torys and the changing worlds of American childhood*, Harvard University Press.
- Baecker, R., I. Small and R. Mandler (1991). Bringing Icons to Life. CHI '91, New Orleans, Louisiana, ACM Press.
- Bruner, J., (1967) *Studies in Cognitive Growth*. Wiley.
- Copeland, R.W., (1979) *How Children Learn Mathematics: Teaching Implications of Piaget's Research*
- Dix, A., Finlay, J., Abowd, G., Beale, R., (2004) *Human Computer Interaction*. Pretice Hall, third edition
- Donaldson, M., (1978) *Children's Minds*. Fontana.
- Ferster, C. B. & Skinner, B. F. (1957) *Schedules of Reinforcement*. Massachusetts: Copley Publishing Group
- Fisher, R., 1995. *Teaching children to learn*, Thornes
- Hammond, J., (2001) *Scaffolding: Teaching and learning in language and literacy education*. Rozelle, Peta
- Hopkins, C., Gifford, S., Pepperell, S., (1996) *Mathematics in the Primary School*. David Fulton Publishers.
- Hughes, M., Desforges, C., Mitchell, C., (2000) Numeracy and Beyond. Open University Press.
- Kyriacou, C., 1997. *Effective Teaching In Schools: Theory and Practice*. 2nd ed. Cheltenham: Stanley Thornes.
- Liebeck, P., (1984) *How Children Learn Mathematics: A Guide for Parents and Teachers*. Penguin Books
- Norman, D.A., (1998) *Design of Everyday Things*. MJT Press
- Piaget, J, (1961) *The Child's Conception of Number*. Routledge.
- Piaget, J. (1970). *Science of Education and the psychology of the child.* New York, Orion Press.
- Piaget, J.(1972) *The psychology of the child*. New York: *Basic Books*
- Polson, M.C., Richardson, J.J., 1988. *Foundations of intelligent tutoring systems*, Erlbaum
- Preece, J., Rogers, Y., Sharp, H., (2002) *Interaction design beyond human-computer interaction*. Wiley
- Richmond, P.G., (1970) *An Introduction to Piaget*. Routledge & Kegan Paul.
- Shneiderman, B., (1998) *Designing the user interface: strategies for effective human-computer interaction.* Addison-Wesley
- Sommerville, I., (2001) *Software Engineering 6th Edition*. Addison Wesley
- Sylva, K. and I. Lunt (1982). *Child Development A First Course*. Oxford, Blackwell.
- Vygotsky, L.S., 1987. *The collected works of L. S. Vygotsky: Volume I: Problems of general psychology*, Plenum Press

VIII.2 Websites

- Nielsen, J., (2001) Heuristics for User Interface Design. [WWW]
  http://www.useit.com/papers/heuristic/heuristic_list.html

- Qualifications and Curriculum Authority (2000) [WWW]
  http://www.qca.org.ukk

- Qualifications and Curriculum Authority (2000) Welcome to the National Curriculum for England. [WWW]
  http://www.nc.uk.net

- Ong, J., Ramachandran, S., 2000. Intelligent Tutoring Systems: The What and the How [online]. Available from
  http://www.dfes.gov.uk/ictinschools/uploads/docarchive/BEC4723%20LAN%20Infrast_19.pdf

- IDC, 2004. Interaction Design and Children [online]. Available from
  http://www.idc2004.org/

VIII.3 Articles

- Anderson J.R., A.T. Corbett, K.R. Koedinger & R. Pelletier, "Cognitive tutors: Lessons learned," J. Learning Sciences, vol. 4, pp. 167–207, 1995.
- André, E., Rist, T., Müller, J., 1998. Integrating reactive and scripted behaviours in a life-like presentation agent. In: Sycara, K., Woolridge, M. (Eds.), Proceedings of the Second International Conference on Autonomous Agents (AGENTS'98). ACM Press, New York, NY, pp. 261–268.
- Arroyo, I; Beck, J.; Schultz, K.; Woolf, B. (1999) Piagetan Psychology in Intelligent Tutoring Systems. In the *Proceedings of the Ninth International Conference on Artificial Intelligence in education*. pp.600-602.
- Beskow, J., McGlashan, S., 1997. Olga—A conversational agent with gestures. In: Proceedings of the IJCAI'97 Workshop on Animated Interface Agents: Making them Intelligent. Nagoya, Japan.
- Bull, S and J Kay, A framework for designing and analysing open learner modelling. 12th International Conference on Artificial Intelligence in Education (AIED 05) Workshop 11, Kay, J, A Lum, D Zapata-Rivera, (eds) pp 81-90. 2005.
- Bull S. and Judy Kay: Student Models that Invite the Learner In: The SMILI Open Learner Modelling Framework. TR 580. January 2006
- Bull, S., Mangat, M., Mabbott, A., Abu Issa, A.S. & Marsh, J. (2005b). Reactions to Inspectable Learner Models: Seven Year Olds to University Students, Proceedings of Workshop on Learner Modelling for Reflection, International Conference on Artificial Intelligence in Education 2005, 1-10.
- Bull, S. & Pain, H. (1995). Did I say what I think I said, and do you agree with me? Inspecting and Questioning the Student Model, in J. Greer (ed), Proceedings of World Conference on Artificial Intelligence in Education, Association for the Advancement of Computing in Education, 501-508.
- Chiasson, S., Gutwin, C. (2005). Design for Children's Technology, HCI-TR-2005-02
- Donker, A., & Markopoulos, P. A comparison of think-aloud, questionnaires and interviews for testing usability with children, In: Faulkner, J.Finlay, F.Detienne, (eds.) "People and Computers XVI – Memorable Yet Invisible, Proceedings HCI 2002", Springer, 305-316.

- Druin, A (1999) Cooperative Inquiry: Developing New Technologies for Children with Children, CHI 99, p.592-599.
- Druin, A., Proceeding of the 2004 conference on Interaction design and children: building a community, June 2004, ACM Press
- Druin, A., and Inkpen, K. When are Personal Technologies for Children? Personal and Ubiquitous Computing v.5, 2001, 191-194.
- Economides, A., Elissaver, G., 2000. Evaluation Factors of Educational Software. IEEE. 113-116.
- Grigoriadou, M., Tsaganou, G. & Cavoura, T. (2003). Dialogue-Based Reflective System for Historical Text Comprehension, Proceedings of Workshop on Learner Modelling for Reflection (Supplemental Proceedings vol 5), International Conference on Artificial Intelligence in Education 2003, 238-247.
- Guilutz, S. & Nielsen, J. (2000). Usability of Websites for children: 70 Design Guidelines. Nielsen Norman Group.
- Gulz, A., M. Haake, Design of animated pedagogical agents- A look at their look, International Journal of Human-Computer Studies 64 336 (2006) 322-339
- Hanna, L., Risden, K., and Alexander, K. 1997. Guidelines for usability testing with children. interactions 4, 5 (Sep. 1997), 9-14
- Jean-Daubias, S. & Eyssautier-Bavay, C. (2005). An Environment Helping Teachers to Track Students' Competencies, Proceedings of Workshop on Learner Modelling for Reflection, International Conference on Artificial Intelligence in Education 2005, 19-22.
- Kay, J., The um toolkit for cooperative user modelling, User Modelling ad User Adapted Interaction 4, 1995, pp149-196
- Kay, J. (1997). Learner Know Thyself: Student Models to Give Learner Control and Responsibility, International Conference on Computers in Education, AACE, 17-24.
- Lester, J., Towns, S., Callaway, C., Voerman, J., Fitzgerald, P., 2000. Deictic and emotive communication in animated pedagogical agents. In: Cassell, J., Sullivan, J., Prevost, S., Churchill, E. (Eds.), Embodied Conversational Agents. MIT Press, Cambridge, MA, pp. 123–154.
- MacFarlane, S., Pasiali, A., (2005). Adapting the Heuristic Evaluation Method for use With children
- MacFarlane, S.,Sim, G., and Horton, M.(2005) Assessing Usability and Fun in Educational Software, in Proceedings of Interaction Design ans Children 2005, Boulder CO, USA, ACM Press
- Malone, T.W., Lepper, M.R., 1987. Making learning fun: a taxonomy of intrinsic motivations for learning. In: Snow, R.E., Farr, M.J. (Eds.), Aptitude, Learning and Instruction. Volume 3: Conative and Affective Process Analysis, Lawrence Erlbaum, Hillsdale, NJ, pp. 223–253.
- Markopoulos, P, M. Bekker (2003) Interaction Design and Children. Interacting with Computers, vol 15, pp. 141-149
- Mazzone, E., Read, J. (2005), Not just bits of paper – design sessions with children, Journal of Department of Computing, Issue n.4, University of Central Lancashire, Preston, UK.
- Moreno, R., Mayer, R., Spires, H., Lester, J., 2001. The case for social agency in computer-based teaching: do students learn more deeply when they interact with animated pedagogical agents? Cognition and Instruction 19, 177–213.
- Mühlenbrock, M., Tewissen, F. & Hoppe, H.U. (1998). A Framework System for Intelligent Support in Open Distributed Learning Environments, International Journal of Artificial Intelligence in Education 9(3-4),256-274.
- Norma, T., Badler, N., 1997. A virtual human presenter. In: Proceedings of the IJCAI'97 Workshop on Animated Interface Agents: Making them Intelligent. Nagoya, Japan, pp. 45–51.
- Oivatt, S., Adams, B., 2000. Designing and evaluating conversational interfaces with animated characters. In: Cassell, J., Sullivan, J., Prevost, S., Churchill, E. (Eds.), Embodied Conversational Agents. MIT Press, Cambridge, MA, pp. 319–345.
- Rickel, J., Johnson, L., 2000. Task-oriented collaboration with embodied agents in virtual worlds. In: Cassell, J., Sullivan, J., Prevost, S., Churchill, E. (Eds.), Embodied Conversational Agents. MIT Press, Cambridge, MA, pp. 95–122.

- Rickenberg, R., Reeves, B., 2000. The effects of animated characters on anxiety, task performance, and evaluations of user interfaces. In: Proceedings of the CHI 2000 Conference on Human Factors in Computing Systems (CHI'00). ACM Press, New York, NY, pp. 49–56.
- Rode, J. A., Stringer, M., Toye, E. F., Simpson, A. R., and Blackwell, A. F. 2003. Curriculum-focused design. In Proceeding of the 2003 Conference on interaction Design and Children (Preston, England, July 01 - 03, 2003). S. MacFarlane, T. Nicol, J. Read, and L. Snape, Eds. IDC '03. ACM Press, New York, NY, 119-126.
- Rueda, U., Larrañaga, M., Ferrero, B., Arruarte, A. & Elorriaga, J.A. (2003). Study of Graphical Issues in a Tool for Dynamically Visualising Student Models, Proceedings of Workshop on Learner Modelling for Reflection (Supplemental Proceedings vol 5), International Conference on Artificial Intelligence in Education 2003, 268-277.
- Straker CJ, Gianinazzi-Pearson V, Gianinazzi S, Cleyet-Marel J-C, Bousquet N (1989) Electrophoretic immunological studies on acid phosphatase from a mycorrhizal fungus of Erica hispidula L. New Phytol 111:215–221
- Takeuchi, A., Naito, T., 1995. Situated facial displays: towards social interaction. In: Katz, I., Mack, R., Marks, L., Rosson, M., Nielsen, J. (Eds.), Human Factors in Computing Systems: CHI'95 Conference Proceedings. ACM Press, New York, NY, pp. 450–455. Taylor, I., McInnes, F., Love, S., Foster, J., Jack, M., 1998.
- Uden L. and A. Dix (2000), Lifelong Learning for Software Engineers, IFIP World Computer Congress - ICEUT2000 Educational Uses of Communication and Information Technologies, Beijing, August 21-25, 2000.
- Walker, J., Sproull, L., Subramani, R., 1994. Using a human face in an interface. In: Adelson, B., Dumais, S., Olson, J. (Eds.), Proceedings of the CHI 1994 Conference on Human Factors in Computing Systems (CHI'94). ACM Press, New York, NY, pp. 85–91.
- Weber, G. & Brusilovsky, P. (2001). ELM-ART: An Adaptive Versatile System for Web-Based Instruction, International Journal of Artificial Intelligence in Education 12(4), 351-384.
- Wyeth P. and G. Wyeth, Electronic Blocks: Tangible Programming Elements for Preschoolers. Proceedings of Human-Computer Interaction - INTERACT'01, Michitake Hirose (Ed.), IOS Press, 2001, 496-503.
- Yacef, K. (2005). The Logic-ITA in the classroom: a medium scale experiment, International Journal of Artificial Intelligence in Education 15(1), 41-60.
- Zapata-Rivera, J-D. & Greer, J.E. (2001). Externalising Learner Modelling Representations, Proceedings of Workshop on External Representations of AIED, International Conference on Artificial Intelligence in Education 2001, 71-76
- Zapata-Rivera, J-D., Underwood, J.S. & Bauer, M. (2005). Advanced Reporting Systems in Assessment Environments, Proceedings of Workshop on Learner Modelling for Reflection, International Conference on Artificial Intelligence in Education 2005, 23-31.

VIII.4 National Curriculum References

- DFEE, Department for Education and Employment, (1999):
  - The National Numeracy Strategy: Framework for teaching mathematics.
  - Curriculum guidance for the foundation stage
- DFEE, Department for Educational Employment, 1999. The National Curriculum for England: Mathematics. London: DFEE.
- DFES, Department for Education and Skills, 2001. SEN Toolkit: Section 11: The role of Social Services. Nottinghamshire: DFES
- DFES. Ofsted, 2002. ICT in Schools: Effect of Government Initiatives Progress Report. London: Ofsted
- DFES, Department for Education and Skills, 2003. The big pICTure.

IX. Appendixes