Department of
Computer Science

UNIVERSITY OF
BATH

Technical Report

Undergraduate Dissertation: Computer-based Mathematics Tuition in Foundation Stage Education: A Prototype Solution

Gregory James Tett

Technical Report: 2004-08 May 2004

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

Abstract

This development project presents an approach to the task of providing a mathematics teaching support tool for use in the foundation stage classroom. The importance of teaching theory and cognitive structures are considered in this context and pertinent issues surrounding the use of technology are also investigated to research the applicability of computer-based learning in education. The involvement of important stakeholders relevant to the domain is maintained throughout to provide additional information and expertise on the methods that should be applied to suitably support the tasks intended. The knowledge gained from all research activities is focussed in the design and implementation of a prototype solution that aims to improve upon current solutions, but also provide the means for evaluating its learning prowess.

Acknowledgements

I would like to thank my project supervisor, Hilary Johnson for her on going support, help and invaluable advice throughout the project. I would also like to thank Lucy Carruthers for her enthusiasm and seemingly endless patience! Moreover, I would like to acknowledge all the staff at North Baddesley Infant School for taking the taking and effort to assist me with my studies.

Project Details

Author:	Gregory James Tett (ma0gjt@bath.ac.uk)
Supervisor:	Hilary Johnson (H.Johnson@bath.ac.uk)
Title:	Computer-based Mathematics Tuition in Foundation Stage Education: A Prototype Solution
Timescale:	October 2003 – May 2004
Deadline:	13th May 2004

Table of Contents

1 Aims and Objectives

1.1 Introduction

Teaching children the very basics of mathematics at the foundation stage of education would appear an easy task, as seemingly the content of the lessons and the concepts covered are very simple. However, a problem found when teaching children at this level is that many have difficulty comprehending the very basics of mathematics, due to its abstract nature. As Jean Piaget describes, mathematics is a difficult subject to learn and indeed it is difficult to teach, primarily because it depends on a child grasping concepts that have no clear relation to anything that they have encountered before (Copeland, 1979).

Unlike any other subject, the mathematics learning experience can consist almost exclusively of working through a series of textbooks where a wider range of concepts and ideas are introduced as children become more competent at problem solving. However, since the 1970s many government led schemes have been introduced with the aim of expanding the 'arithmetic diet' through a variety of practical teaching methods. The most recent and perhaps important of these schemes is the DFEE's national numeracy framework, launched in 1999. This has introduced many innovative and exciting 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. Through these teaching methods children are more motivated to learn and can more readily become comfortable with mathematical and numerical concepts which form the basis for learning throughout their education.

However a major problem 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. Being the first year of education there are also a vast range of ability levels which results in children taking different amounts of time to become entirely comfortable with the concepts they are taught. In summary, individual tuition for all children is almost impossible.

Information and Communications Technology (ICT) has become an interesting and important educational tool for attempting to alleviate or even solve this problem. Computer-based content-rich tutorials that provide a relevant mathematical challenge, whilst allowing children to gain experience of human-computer interaction have clear educational value. However given the user group, special consideration must be given to the nature of the interaction and the modes for communication, which existing solutions seem not to provide in suitable detail. Therefore their educational worth cannot yet be appropriately evaluated with a view to offering a solution to the original problem.

Hence there is a vast scope for incorporating the objectives of mathematics education within computer-based media whereby appropriate research into how to make a tutoring system truly usable may hold the key to supplementing mathematics education.

1.2 Overall Aim

The overall aim of the project is to produce a prototype solution to the mathematics tutoring problem that complies with the objectives of the foundation stage curriculum. The prototype should support learning and teaching processes and be an effective mathematics teaching support tool for children at the foundation stage.

1.3 Key Objectives

The key objectives of the project to be undertaken, to achieve the overall aim are:

To research all relevant theoretical areas of the problem domain and thus gain a better understanding of the problem and its related topics. This should include a detailed assessment of the national curriculum so that we are able to examine the scope of the mathematics education

To investigate and understand relevant human-computer interaction (HCI) research that relates to designing software aimed at children, and suggest ways in which this may be applied in order to benefit the system's usability.

To gain an understanding of the practical issues that relate to the problem. This should involve observing educational activities of the foundation stage classroom as well as the methods teachers use to impart knowledge and practice important mathematical concepts.

To undertake empirical studies that will involve interviewing teachers, the education authority and developers of existing tutoring systems and thus gain an exposure to the issues that must be overcome if the system is to be successful in the classroom environment. Accordingly we should identify how to design a system that will be accepted by the children but also provides a challenge.

To critically evaluate current tutoring solutions and thus identify how they support and consolidate the concepts taught in the classroom. Also to assess how their well-designed features can be applied to the design of the prototype solution.

To develop a highly interactive prototype solution, in accordance with the overall aim of the project, that can be easily extended to cover all mathematical concepts taught in the national curriculum.

To deploy and test the prototype in the foundation stage classroom and evaluate whether it is successful at supporting the development and consolidation of mathematical knowledge.

To consider how future technologies will impact the problem domain and how these technologies may be utilised to aid the usability of the system or its tutoring prowess.

2 Literature review

2.1 Introduction

Learning; "the act, process, or experience of gaining knowledge or skill", and teaching; "imparting knowledge or skill" would seem to go hand-in-hand in a fairly simple fashion. However to understand what these terms truly mean and how to apply each of them in the context of a mathematics tutoring system for children at the early year's level requires knowledge of a wide range of subjects.

This literature review provides an investigation into this problem domain, by investigating how children learn and how they are taught mathematics. It discusses the relevance of computer-based tutoring and human-computer-interaction in this context, with reference to many sources to draw together important research and ideas.

The aim of the literature review is to identify particular issues of educational activity and psychological learning that impact the scope of the problem and provide the basis for the implementation of a solution that effectively supports the tasks and goals of primary age children.

2.2 How Children Learn

In order to learn more about the psychology of learning it is necessary to investigate theories of how children learn.

2.2.1 Development of the mind

At different ages, children's relation to their surroundings and mental development varies in reflection to their changing interests, contexts and characteristics. This has been an area of extensive study since the early 1900s, driven by the desire to find the most efficient and effective ways for children to learn and teachers to teach. Some of these theories offer invaluable information about the child's mind which forms an important underpinning to the design of an educational tutoring tool.

Jean Piaget is considered to be the psychologist to have contributed most to the field of children's intellectual development. As Liebeck (1984) explains, Piaget's theory of cognitive development states that children pass through a sequence of developmental stages through which they attain conceptual knowledge of number conservation[1], length conservation, and other mathematical concepts, in a well defined order. Piaget claims that the *order* of these stages is the same for all children, as each stage builds upon the knowledge of its predecessor. However, Piaget also states the speed of movement between these stages is certainly not the same for each child, hence his summation of four broad periods that describe cognitive development:

- Sensori-motor period: (Birth to 18 months)
  In this period the child's behaviour evolves from spontaneous movement to reflexive movement and acquired habits. The child develops from an initial egocentric viewpoint that there is no distinction between himself and the rest of the world, to an

[1] Conservation is described as the point of understanding at which a child fully grasps the meaning of a mathematical concept and can apply it in many abstract circumstances, without support.

understanding that the world consists of people and objects that are independent of him and his actions.

- Intuitive period: (18 months to 7 years)
  Piaget claims that this period comprises of two sub-stages; 18 months to 4 years and 4 years to 7 years. Throughout this period Piaget suggests that the child prepares for 'concrete operations' which are established and consolidated in the next period.
  From 18 months to 4 years the child begins to gain an understanding of representation, characterised by a growth in his ability to describe objects and actions. Towards the end of the period the child has conservation of number, and can also begin to build conceptual knowledge of mathematical activities and concepts.
  From 4 years to 7 years the child gradually makes the transition from pre-operational thought to operational thought, embodied through the realisation that their viewpoint on the world is not necessarily the only viewpoint. The child has not yet fully developed reasoning skills and thus will solve problems by empirical testing rather than logical reasoning. The child is also dominated by their perceptions, and thus is likely to be misled by what they see.
- Concrete operational period: (7 to 12 years)
  The child can begin to apply logic to physical situations in real or imaginary situations through operational thought. Piaget proposes that the child has an emerging understanding of mathematical concepts, and therefore does not rely on perception as much as in the intuitive period. The ability to recognise that a problem can be solved and then work backwards, reversing their previous train of thought, is known as reversibility and Piaget claims a child can apply this concept in a variety of situations.
- Formal operation period: (12 years and above)
  The child has the ability to reason and hypothesise and make deductions on the basis of logical thought, without the need for physical objects as the basis for understanding.

An important area of Piaget's research that is particularly relevant to this project is his theory that children of the intuitive period gradually become 'conservers' of number. As Hughes (1986) explains, Piaget conducted the class-inclusion experiment to test whether children had true conservation of number, by testing whether they understood the terms 'two', 'three', and so on. However Piaget argued that a child will not fully understand these terms until they understand that the set 'eight' (for example) contains two subsets, 'two' and 'six', and by reversibility, the two sets 'two' and 'six' form the set 'eight'. He claims this knowledge is the foundation for understanding addition and subtraction, which occurs only when the child is thinking operatively, at the age of seven. The role of a computer-based tutor in helping to support this process may be to present information in different ways and in different contexts that help the child to make the necessary connections between numbers and sets of numbers, such that they can make this inference. Presentation of the same concept in multiple ways could help the child to extract new or enhanced meaning of number and class-inclusion, providing the basis for understanding addition and subtraction.

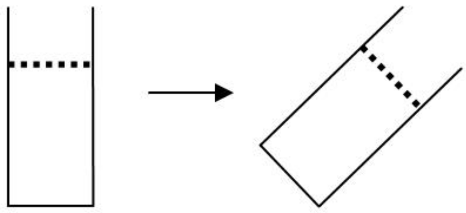
2.2.2 Inference

In the intuitive period, Piaget claims that perception is the overriding factor in affecting a child's interpretation of a mathematical problem, and they do not think operationally or logically until the age of seven (approximately). This is an important premise, which is reinforced by Markopoulos (2003) who states that children at this age have not yet fully developed reasoning skills and rely on perception to 'fill-in the gaps'.

However research from Liebeck (1984) and Bryant et al (1971) would suggest that children do in fact use logical reasoning earlier than Piaget and Markopoulos claim. Liebeck explains that in an experiment with 6 year olds, the children could make the inference that `A = B, B = C`, therefore `A = C` when solving simple equations over addition. Similarly, Bryant et al found that 4 year old children could make the inference `A > B, B > C`, therefore `A > C` when questioned about the lengths of a set of coloured rods. These experiments led Liebeck and Bryant to conclude that children apply inferences rather than egocentric perception when problem solving.

In a further experiment Bryant et al asked 4 year old children to draw the water level on a container, whilst the container was sat on a table. The children understood that in this example the water level is always parallel to the base of the container. However when posed a similar problem where the bottle is tilted and they must draw the line on the container, Bryant et al observed the following results:

Figure 2.1 Bryant et al (1971) found that 4 year old children made the inference that water level is parallel to the bottom of the container in the familiar case, but when tilted to form an 'unfamiliar case' they inferred that the water level would still be parallel.



This experiment shows that the children made the inference that they should apply the knowledge they had learnt in the familiar case, in the unfamiliar case. Bryant et al gives the reasoning that the child perceived the problem, and made the connection with their previous experience that the water level in a container is parallel to its base. This is known in developmental psychology literature as 'overextension of a rule' and is indicative of children at the intuitive stage of development.

Given these findings, it is possible to recognise that children use perception as well as inference when problem solving, thus combining the findings of Piaget, Markopoulos, Liebeck and Bryant et al. Therefore we can recognise that a fundamental part of problem solving in the intuitive period relies on children perceiving a situation and making connections to their existing knowledge. As Richmond (1970) states "All new experiences must be related to experiences which the child already understands." In accordance with this we can see that the role of a computer-based tutor is to support the exploration of unfamiliar problem spaces, where children can apply their knowledge and make new connections to their experiences. Multimedia software is suited to this task as through animation, manipulation and reorganisation it is possible to demonstrate how materials evolve and change in certain situations[2]. Clearly this can be easier for a child to interpret than with static learning materials.

2.2.3 Interpretation

Through her research into the interpretative abilities of children, Margaret Donaldson (1978) claims that children do not always interpret language in the same way as adults. Donaldson discovered that children in the intuitive period of development interpret the gist of what they are taught and have a fundamental urge to make sense of it. However, Donaldson found that children rely on language as only one of many cues to assess the intentions of the speaker.

Donaldson found that when the language being used, or concept being described is unfamiliar, the child relies on the non-linguistic behaviour of the speaker to interpret meaning. Primarily this is because the child's knowledge of language is not as developed as their understanding of non-linguistic communication, where they are "on surer ground". Importantly she found that if the child understands the context of the situation being discussed they are more likely to

[2] For example the change in water level when a container is tilted.

interpret the meaning of what they are told more precisely. It is for this reason that both Donaldson and Liebeck (1984) agree that abstract mathematical concepts, prone to misunderstanding, should be taught in the context of real-life, physical experiences.

Vygotsky supports the views of Liebeck and Donaldson in this context. He emphasised the need for children to learn by having “authentic situations in which they must resolve dilemmas”. In his zone of proximal development theory, Vygotsky stated that children do not possess the ability to operate on a completely abstract level, and therefore require additional teaching and learning activities centred on some concrete example to begin understanding concepts. This is an important finding because if we can make the context or theme of the computer-based tutorial accessible to children in order for them to make connections to their previous experiences, it may be possible to reduce learning time and thus make the system intuitively easier to use.

Clearly then static information presented in the form of independent and hypothetical entities has too many barriers to entry for children to understand as it lends itself to easy misinterpretation. However, if the information presented is understood to relate to situations in real-life contexts it ceases to be static, and can be applied with greater success, as we have discussed. Enabling children to relate to problems quickly, and easily make links with their past experiences in new situations is an important aim for the computer-based tutor. In this context it is possible to recognise that the use of relevant metaphors or examples could help to reduce learning time and create an intuitive learning environment.

2.2.4 Human Memory

Norman (1998) performed a number of detailed studies into human memory to discover the best methods to support remembering and understanding. He revealed that memory has three categories:

- Memory for arbitrary things: the items to remember have no particular meaning. They have no clear relation to each other or things already known.
- Memory for meaningful relationships: the items have meaningful relationships with each other or with things already known.
- Memory through explanation: the material does not have to be remembered, but rather can be derived from some explanatory mechanism.

Memory for arbitrary things is the simple remembering of what is to be done, without understanding why. It is possible to make the connect this with how we learn truly arbitrary things like the alphabet, but it is argued that it is not really possible to learn abstract associations or sequences in this way. Norman explained that people attempt to form mental structures in order to remember arbitrary things, which are usually unsatisfactory and result in poor learning “…it is not the best way to go, not if there is any choice in the matter” (Norman, 1998). It is for this reason that we use a tune to remember the alphabet for example, as the natural constraints of rhyme and rhythm provide an appropriate structure that has the effect of simplifying memory tasks.

At the foundation stage it can be inferred that much of the mathematics material will be viewed as arbitrary to children in that their previous experiences will have no clear relation to the concepts being taught. However, we can recognise that it is of great importance that foundation stage lessons are understood in depth as they form the basis for future education. Therefore it is essential that learning has a sensible and meaningful structure, so that relationships can be drawn between the items to remember and existing knowledge; “when things make sense, they correspond to knowledge that we already have [enabling] new material to be understood, interpreted, and integrated with previously acquired material.” Norman’s explanation adds weight to the previous discussion concerning the use of appropriate metaphors for learning that children find interesting and accessible. Also it is clear that where mathematical abstractions are presented and learnt in accordance with these rules we can integrate remembrance and understanding more efficiently and easily.

2.2.5 Accelerating development

Piaget's research found that learning is subordinate to development, meaning that no amount of teaching can increase the rate at which a child passes through the stages of cognitive development. However further psychological research has challenged this theory and concluded that the influence of teaching can accelerate the developmental progress.

Bruner (1967) conducted a 'conservation of water' experiment to test whether children between 5 and 7 years could predict the water level in container Y, which is hidden behind a screen, when poured from an identical water container X. Bruner conducted a first experiment to test whether the children could apply the knowledge they had gained from observing the 'familiar' state[3], in the 'unfamiliar' state, without the aid of perception. He then conducted a second experiment to test whether the children could apply their knowledge appropriately after participating in a period of additional teaching and support. In this period the teacher discussed the problem in more depth, analysed the answers that had previously been given and gave constructive feedback, and posed a number of problems where the children could apply their knowledge in different scenarios. The aim of this period of teaching was to reform misconceptions and develop the confidence of the children at applying their knowledge base, so to reduce their reliance on perception. Bruner noted the following results:

Figure 2.2 – Results from Bruner's (1967) experiment into the effectiveness of teaching on influencing the rate at which children develop an understanding of conservation of capacity

		Age of Children		
		5	6	7
Percentage of children comfortable with conservation of water	Before teaching	20	50	50
	After teaching	75	90	90

It is possible to infer from these findings that an initial session of teaching does not necessarily provide a firm basis of understanding, especially with children in the intuitive period of cognitive development. The experiment showed that further teaching, discussion of the problem and practice applying the concept in real-world situations provided a deeper level of understanding. These findings link with those of Liebeck (1984), and Bryant et al (1971) in that they infer that inaccuracies when applying knowledge is the main cause of mistakes of children in the intuitive period, which can be positively affected through extra teaching and practice.

A computer-based tutor could be a good tool for supporting the application of mathematical concepts in a variety of situations. Being a piece of software, one of the main advantages of the tutor is that it can provide unlimited opportunities for practice. Concepts that are complex can be demonstrated in front of the user using animation, and also a number of problems can be posed to test the child's knowledge throughout the learning session. This interaction may provide the learner with a different viewpoint on the concept allowing new connections to be made. The tutor could therefore be used as the 'additional teaching support' described in Bruner's experiment, providing opportunities for reinforcing ideas and applying concepts.

[3] The familiar state being the water level of container X, before being poured into container Y.

2.2.6 Application of Concepts

As already discussed, developing connections and associations between abstract concepts and experiences in the real world, forms the basis for true understanding in children. The National Numeracy Strategy (DFEE 1999) also details the need for learning to be expressed in real-world contexts so that teaching methods are clearly understood, and knowledge is assimilated efficiently.

Hughes et al (2000) describes learning as acquiring links or connections to real world experiences and having them at ready disposal for application in different contexts. He says that prior to the development of logical thought in the operational stage of development, "successful performance depends on the degree of familiarity of the elements of the problem with associations already held in the mind." He views the opportunity to practice applying concepts in a variety of situations as fundamental to understanding. Furthermore, Hughes explains that "[children] need to learn how to use their existing knowledge when they are confronted by new problems in novel contexts. There is little point in being 'numerate' if they cannot apply what they know."

As we have seen, Vygotsky, Donaldson (1978) and Liebeck (1984) agree with this theory. They state that learning metaphors and authentic examples can help to form the required connections between abstract concepts and the real world. Donaldson explains that learning metaphors give children a "concrete focus" that they can more easily relate to without needing to think through the abstract mathematical concepts they represent. An example of this is a story, where children can relate to the characters and can develop an interest in the plot line, without the need to consider the educational side of the lesson. Donaldson states that it is this that allows children to become more familiar with mathematics without the need for logical reasoning.

Using a computer-based tutor as a means for conducting these 'real-world' examples could be less time-consuming and more practical given the class size and spectrum of mathematical competencies. Through thoughtful use of animation, graphics and sound it may be possible to allow children to explore ideas in more intriguing situations, where they are able to establish a non-abstract focus on the learning session. Providing opportunities for practice in such contexts could enable the children to develop a fuller understanding of new mathematical concepts. However, we also must consider that when children reach school age they will already have developed their own ideas and assumptions about the world. If the contexts posed for learning do not agree with these the children may become confused and thus the attempts to expand the child's scope of learning will fail. Therefore it is important that the target age-group are well understood so that appropriate scenarios for questioning can be posed.

2.3 Mathematics

Mathematics is a vitally important subject in the national curriculum. Liebeck (1984) explains that mathematics education helps children to develop the ability to communicate in an unambiguous and concise manner and also to develop the ability to reason logically. She states that mathematics is of particular importance in a child's education because of its direct application in a variety of situations in everyday life. Furthermore mathematics provides an intellectual challenge for children, with an inherent "…enjoyment and aesthetic satisfaction from exploring this area of human knowledge" (Hopkins et al, 1996).

Mathematical development in the foundation stage depends on a child becoming confident and competent in key skills such as counting, sorting, matching, seeking patterns, and making connections with numbers, shapes, space and measures. The teaching techniques encouraged through the national numeracy strategy aim to develop mathematical skills through supportive teaching activities such as imaginative play, stories, songs and games. An important part of this learning interaction is to develop the child's communication skills and ability to use mathematical language correctly, with the underpinning of 'sound' mathematical knowledge (Hitcham, 2000). This has a particular relevance to the use of computers to support mathematics teaching. A computer-based tutor can communicate through its interface using sound and text to convey ideas and concepts. This interaction can add an extra dimension to the learning process by helping children to focus their thinking and practice using mathematical language appropriately.

Piaget states that the most important and difficult part of a child's mathematics education is constructing the fundamental conceptual basis for mathematical thinking. However Hopkins et al describe that "…it is of little use to a child if they can correctly complete pages and pages of sums but doesn't know how to get started when faced with an unfamiliar problem or task presented in a practical rather than a written form" (Hopkins et al, 1996). Thus the key to developing mathematical knowledge is to form a base of understanding and to provide the means for applying it, which clearly builds upon previous discussions.

Hopkins et al describe three key skills that are needed for children to effectively tackle unfamiliar processes in mathematics:

- Mathematical decision making, the ability to notice that a problem is not just allied to one method of solution, and making links between concepts.
- Communication, the ability of a child to articulate their thinking and to move towards using more abstract language, symbols and representations.
- Reasoning mathematically, the ability to generalise mathematical relationships and think through a problem logically. This also involves developing the ability to move from concrete experiences for mathematics application, to abstract, and back again.

It is possible to see that a computer-based tutor is ideally suited to providing support for the first two of these three skills, with respect to the cognitive skills of children in the intuitive period. As previously discussed, communicating ideas and giving instructions through sound and text at the interface could help the child to learn how to use mathematical language appropriately and correctly. It could also provide the basis for children to 'copy' the tutor when developing this skill. Furthermore, thoughtful scenarios for applying their knowledge could enable the development of decision making skills and thus the ability to interlink mathematical concepts and deepen their understanding. Mathematical reasoning, Hopkins third key skill for mathematics, is considered the most difficult to acquire. Piaget's theory of cognitive development states that children at the early year's level do not have the ability to reason logically and thus are dominated by their perception of given situations to arrive at solutions. However, as Bruner (1967) found in his 'conservation of water' experiment, supportive teaching can help to accelerate development, whereby the problem is discussed further and the relevant concepts are applied in a series of scenarios.

As we have seen, one of the key advantages of the computer-based tutor is that it can model a variety of relevant examples and scenarios allowing children to practice applying their mathematical knowledge without the constraints of the budgetary or safety requirements of the classroom. Therefore, in accordance with Bruner's findings, we can recognise that such practice may help to accelerate the child's ability for mathematically reasoning, or at least lay the foundations for acquiring reasoning skills.

2.4 Teaching

"High-quality teaching is oral, interactive and lively. It is not achieved by adopting a simplistic formula of drill and practice or by expecting pupils to teach themselves from books. It is a two-way process in which pupils are expected to play an active part by answering questions… and demonstrating their methods…" DFEE (1999).

Teaching involves exposing children to appropriate material and concepts at pertinent times so that their educational development follows a suitable path. This is referred to as 'scaffolding', which Hammond (2000) describes as "…support that is designed to provide the assistance necessary to enable learners to accomplish and develop understandings that they would not quite be able to manage on their own." Furthermore, Hammond states that a teacher must withdraw this support gradually as the learner becomes increasingly able to complete the task alone, at which point they should develop "new tasks and new learning that [are] an extension of previous understandings." In summary, teaching is about support, consolidation and extension of knowledge. We can relate this to the tutoring system such that a lesson should build a base of understanding and then incrementally increase its level of difficulty in accordance with the child's increased understanding. This clearly provides the basis for expanding the child's knowledge of the concept.

Evidently the sentiments of "high-quality teaching" and the principles of scaffolding are directly applicable to the design of a computer-based mathematics tutor. The tutor can be considered as a 'teacher of sorts' because it supports learning and promotes understanding in a complementary way to classroom teaching. However we must consider that all aspects of teaching are not necessarily programmable. A large part of teaching is borne out of experience, where a teacher must consider how to present information and how to invoke thought and understanding in appropriate ways. Teachers must also ensure that the educational path closely follows the objectives and standards of the national curriculum such that all children reach the appropriate level of numeracy by the end of the foundation year. It is necessary that we gain an understanding of these processes so to identify ways in which the tutor can maximise its potential as a support tool for teaching methods.

2.4.1 The National Numeracy Strategy

The DFEE (1999) launched an initiative in September 1999 that stipulates that all schools provide a daily 1 hour structured mathematics lesson, known as the 'numeracy hour'. During the numeracy hour the DFEE state that the whole class must partake in oral, written and mental work to practice using their mathematical knowledge.

The aim of this strategy is for 75% of all 11 year olds to achieve at least level 4 in the National Curriculum tests for mathematics. It was introduced to help schools teaching primary age pupils to set appropriately high expectations for their children and understand how they should progress through the primary years. The notion that "[children] need regular, planned opportunities to develop their mathematical knowledge" (DFEE, 1999), is the driving force behind the numeracy strategy, which includes a well structured teaching framework and lesson ideas. The strategy has led to widespread changes in the way mathematics is taught throughout England and Wales.

The DFEE's framework for teaching mathematics recommends that mathematics teaching is based on four key principles:

- Dedicated mathematics lessons everyday.
- Direct teaching and interactive oral work with the whole class, and in groups.
- An emphasis on mental calculation.
- Controlled differentiation, with pupils applying the concepts of mathematics in many situations.

These principles provide a framework for developing mathematical concepts, with the teacher playing a major role. Clearly the first three principles require the teacher to use appropriate teaching techniques to assimilate knowledge whilst catering for the spectrum of mathematical ability in the classroom. However it is possible to recognise that a computer-based tutor could also consolidate these principles. Through the use of multimedia software the tutor can provide interactive scenarios for children to apply their knowledge where mental calculation plays a key role. Through direct manipulation of on-screen objects the tutor can also provide a different perspective on how to arrive at an answer, which may help to reinforce the understanding of classroom taught concepts.

The need for differentiation is an overriding reason for the use of ICT and particularly computers in the classroom. The DFEE promote the use of computers in education because they provide a different medium for interaction and communication, and invoke a willingness to learn. They state that the human-computer interaction is different from any other mode of communication in the classroom and as such provides an interesting and motivating method for children to learn. Consequently, where we can support mathematics learning, and simultaneously develop a child's computer related skill-base it is clear that the learning session is both efficient and valuable.

The DFEE suggest that "good, direct teaching" of mathematics is achieved through a balance of:

- Directing: sharing teaching objectives with the class at the start of the learning session, and ensuring that pupils know what they are expected to do.
- Instructing: giving information in a structured manner such that misinterpretation is avoided.
- Demonstrating: showing, describing and modelling mathematics with appropriate resources and visual displays.
- Explaining and illustrating: providing accurate explanations that draw on real-world situations. The teacher should also explain new concepts and walk-through the thought process required to arrive at the solution.
- Questioning and Discussing: involve the whole class by posing questions that demand that children apply their knowledge. The teacher should also analyse and discuss answers to their questions and give constructive responses to take learning forward.
- Consolidating: reinforcing and developing the concepts that have been taught with a variety of classroom activities.
- Evaluating Pupils' Responses: identifying mistakes and using them as positive teaching points by talking about the misconceptions that led to them.
- Summarising: reviewing what has been taught at the end of the session and correcting misunderstandings. Also picking out key ideas and concepts and making links to suggest how these might be applied in real-world contexts.

The principles of "good, direct teaching" describe the aspects of the teacher-child interaction such that effective assimilation of knowledge is achieved. The DFEE state that it is the job of the teacher to introduce new concepts, and instruct, demonstrate and explain these appropriately, and thus clearly it is not the job of a computer-based tutor to support all of these tasks. Clearly though it can provide the differentiation for children to apply and consolidate their knowledge, and thus become comfortable at using these concepts. The computer-based tutor may also be possible to apply one or more of these teaching techniques to instantiate a learning session that is appropriate given the aims of the national curriculum and is also familiar for the children, with a view to helping children become comfortable with a particular learning style.

2.4.2 Teaching Techniques

The efficiency and appropriateness of teaching techniques is a very well studied area. There is a constant desire to find new techniques that facilitate learning, engage and motivate children and efficiently assimilate knowledge and skills, with the goal of raising the standard of numeracy in schools.

There are many approaches to teaching children, each of which aims to provoke different responses and nurture different modes of thinking within children. The DFEE (1999) state that from the principles of "good, direct teaching" we have seen that questioning is an important part of all mathematics lessons. Questions can be used by teachers in many forms. To challenge a child to recall information, teachers pose questions that require a quick response to test whether they have understood a concept and can recall it automatically. An example might be "1, 2, 3… what is next in the sequence?" For children in early year's classes this is well within their base of knowledge, but aims to build confidence and maintain a consistent level of child participation in the lesson. Questions can also be used when building on previous knowledge to introduce new ideas. For example if the child answers "4" to the first question, the teacher might then say "what number is 2 less than 4?" Reasoning questions such as these require the child to apply their knowledge of the number sequence in a different way from simply counting upwards. This requires the child to think through an idea and possibly apply their knowledge abstractly, which assesses how 'deeply' they have understood the mathematics topic. The DFEE (1999) state that responses given to reasoning questions provide even more scope for developing knowledge as constructive responses from the teacher can clarify ideas and consolidate important concepts.

The tutor is ideally suited to posing questions that require a child to recall knowledge or think through some situation. By their nature questions between a child and a teacher require a one-to-one interaction which can be effectively simulated by a child and a computer. The tutor can pose questions of various difficulties and provide feedback to the child on whether they are right or wrong. If the child has answered wrongly, it can also display the right answer which may help to resolve the child's mistake. However, we must consider that where the child has answered a question wrongly because of some misinterpretation, the tutor has no means of providing additional explanation or support. In this situation we can see that the computer based tutor is quite inflexible.

Another teaching style that forms an important part of the national numeracy strategy is that of group work. When all children are involved in group activities the teacher can readily adapt the mode of instruction and explain concepts in different ways to aid clarity and resolve misconceptions. They can also use visual props and other resources that provide "…opportunities for learning in contexts which children find relevant and interesting" (DFEE, 1999). Group work also provides the opportunity for children to learn from each other. Children can correct each other and copy each other, which can help them to grasp concepts more efficiently. The value of this is described by Vygotsky's claim that "children undergo quite profound changes in their understanding by engaging in joint activity or conversation with other people".

The advantages offered by group working can be related to the tutoring system through interface personas. Interface personas provide working companions for users and can provide instructions and feedback. Where children can work alongside a persona they can copy its actions and its use of mathematical language for example, providing similar benefits to group work in terms of 'learning from each other'. A persona also provides a figure for the child to relate to that they could potentially find interesting and endearing.

Themed lessons are used extensively by teachers to assimilate knowledge in relevant contexts. The DFEE stress the importance of using appropriate metaphors and examples so to make the learning environment more efficient, enjoyable and successful, which is reinforced by Cooper et al (2003) "…[metaphors] are a vital means of extending a children's thinking and fostering their knowledge and attitudes". As already established, themed lessons provide an accessible focus for children that does not require them to think in abstract terms. This can help to make the necessary association between a mathematical abstraction and a practical application which might otherwise be too difficult.

Themes can be simple as “I have 6 sweets, if I eat one, and give one [to a child], how many do I have left”. This represents the mathematical abstraction “6-2” which ordinary could be confusing for a child to understand, especially as statically presented information. Other examples that have a firm mathematical basis might be:

- Games of ‘snakes and ladders’ or solving specially designing puzzles, requiring children to apply their knowledge of the number sequence and counting.
- ‘Playtime’ where the children run a shop, requiring the use of mathematical skills in a real-world context.

Clearly physical teaching activities cannot be reproduced in a computer-based interface, but they can be simulated. With the use of graphics and animation it is possible to provide a wide scope for mathematics application, which could help to reaffirm ideas and complement the activities of the classroom. Such applications could cover fantastical or far ranging scenarios that interest the children in a more practical way than the equivalent group activity in the classroom.

2.4.3 Motivation

Motivation of the learner is of prime concern to all teachers. It is motivation that sustains the learner’s attention and encourages the learner to strive to achieve their goals or objectives.

Copeland (1979) explains that motivation comes from a) the child and b) from the environment. He also 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. Thus teaching has to be relevant to the learner so that it expands upon their existing knowledge and set of experiences. Piaget reinforces this theory. He claims that a one-year old child is not motivated by reading, because their interpretation of letters is beyond their present schema, “the features of the situation are so different from other situations for which the child has well developed schemas that he is unable to accommodate them” (Copeland, 1979). However a new rattle will stimulate the child’s interest and attention because they understand its purpose and meaning.
Clearly the same theory is applicable for the tutoring system. Children in the early year’s classroom cannot easily absorb information in static form and thus if the tutor were to present pages of sums that demanded the child to make decisions on the basic of logical reasoning this would be well above their schema. It would be vastly more successful and motivating for a child to solve problems set in contexts that they can easily relate to, as promoted by the DFEE in their definition of “high-quality teaching”.

Motivation is essential for efficient learning. Keller (1983) provides the following guidelines which summarise the four main factors necessary to achieve optimal motivation.

- Attention: this needs to be established and maintained through-out the learning session
- Relevance: imparting on the learner the significance and value of what is being taught is motivational. However, as each learner is different it is likely that different situations will be relevant in different ways.
- Confidence: the learner’s motivation is affected by the belief in their ability to achieve their personal goals. Confidence is further enhanced by the learner’s performance in the learning session.
- Satisfaction: this is the reinforcement and consolidation that results from the successful completion of the learning session. Thus the learner should receive feedback on their achievement.

Multimedia software is ideally suited to fulfilling Keller's model and complying with Piaget's guidelines for the following reasons:

Attention

By referring back to Keller (1983), three actions are cited as being important for gaining and sustaining the attention of learners:

- Varying the appearance or sound of the instructional materials
- Using relevant concrete examples of concepts that are presented
- Surprising the learner through novelty and incongruity

Multimedia software is well suited to fulfilling these actions. As we have discussed, we can combine sound, pictures, animation and text to present information in many different ways. This is advantageous because we can provide a variety of concrete examples that offer different viewpoints on how to approach mathematical problems. Moreover, sound can be used in a musical context to provide an extra dimension to themed tasks, as well as an educational context whereby instructions and feedback can help the children to learn how to use mathematical language in appropriate situations.

Reference

Keller's identification of relevance as motivational in the learning session clearly agrees with Piaget's view, such that effective learning is only achieved using appropriate and relevant examples. When we relate this to the computer-based tutor it is apparent that there is a need to consider that examples must closely relate to the activities of the classroom and also the guidelines of the national curriculum so that they are truly motivational and interesting for the children.

Confidence

The need to increase learner confidence is of clear importance with child users. As Keller states, user motivation is positively affected through their increased ability to achieve their tasks. In the tutoring system we can provide lessons that incrementally increase in difficulty allowing users to gain confidence from being able to accomplish basic tasks, before expanding their knowledge in subsequent tasks. As Hammond (2000) described in his scaffolding theory to teaching, incrementing the difficulty of lessons allows children to learn new skills by expanding their existing base of knowledge. Keller adds weight to this premise by noting that this teaching technique also has motivational qualities and thus it is possible to recognise that it can be used to create an efficient learning session in the tutoring system.

Satisfaction

Where a sequence of tasks are presented within a suitable set of relevant contexts, and where they incrementally increase in difficulty so to nurture the user's increased level of understanding, it is possible to notice that satisfaction can be achieved by reinforcing and consolidating the user's knowledge. We can use feedback via sound and animation to reward the child for answering questions correctly, allowing them to gauge the level of their achievement. As Keller describes, it is achieving both short-term and long-term goals that provides satisfaction for users and thus maintains their level of motivation throughout the learning session.

2.5 ICT for Early Years

Information and Communications Technology (ICT) is an important part of the national curriculum for all children under 8 years of age. As described by Cooper et al (2003) "there is a clear expectation on the part of the government that children should use information and communications technology in both their education and adult careers". The reasoning for this is reported by DfES/Becta (2002), "there is evidence to suggest a correlation between 'good ICT skills' and improved attainment."

Children have a natural curiosity and enthusiasm to interact with different mediums, especially technologies. The DFEE propose that multimedia and interactive resources can be particularly motivating at the early year's level because of this enthusiasm for interaction. This is advantageous for the computer-based tutor as gaining the attention of the learner and sustaining it throughout the learning session is important if the session is to be both successful and motivational, as Keller (1983) and Hammond (2000) agree. Hitcham (2000) also reinforces this premise by stating that "a computer program with good quality graphics and animations can provide an exciting context for mathematical thinking, and can be an effective stepping stone between concrete experiences and more abstract ideas".

The DFEE (1999) state that developing ICT skills at the foundation stage involves using a variety of technologies. Video, TV, audiocassettes, computers, and so on, are used as supportive learning tools in the classroom as they can:

- Enhance present learning, with the use of sensitive feedback and dynamic presentation
- Support and extend children's educational development
- Assist children to generalise concepts and skills
- Engage children in self-directed learning

Computer-based multimedia software (as an ICT tool) is ideally suited to fulfilling these objectives. Through meaningful contexts for learning and dynamic presentation of information, it is possible to 'bring situations to life' such that children can apply their mathematical skills effectively, and without the problems associated with static materials. "Learning with ICT can provide added value in extending learning opportunities for children, often in ways only an ICT resource can offer" (DFEE, 1999).

The DFEE specify a set of principles that underpin the ICT framework policy. These principles build upon those set out in the national curriculum for children aged 3-5 years, and specify the need to:

- Understand the different ways in which children learn, and how ICT can be used as an educational tool to support learning. ICT should be embedded into the learning environment and used in contexts that are meaningful to children.
- Recognise that relationships and interactions are at the "heart of all learning experiences". The use of ICT should encourage children to develop shared understandings with other children and adults, as it is proven that they learn more effectively in collaboration.
- Understand that inclusion is promoted through a varied learning environment. ICT provides individual children with motivation and encouragement, and also a distinct differentiation in their learning diet. ICT brings an awareness of technologies that are important part of a child's life both now and in the future.
- Provide access to a range of ICT resources. Teachers should plan how to regularly integrate ICT into the learning environment.

We can directly apply these principles when designing a computer-based tutor in order to develop a valuable learning tool. The statement that children learn more effectively in collaboration can be related to the tutoring system through an interface persona (as already discussed) but also by making the system accessible to more than one child at a time. This would enable children to discuss tasks and concepts and thus benefit from collaborative learning.

We can also recognise that the child-computer interaction is a beneficial learning experience in itself. Children indirectly develop their skills in using and controlling input devices and manipulating on-screen objects at the same time as developing their mathematical knowledge, thus combining two important subjects of the national curriculum.

2.6 Human-Computer Interaction (HCI)

When developing any computer-based technology there is an inherent need to consider and apply the issues of HCI and usability. 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. It is also necessary to consider the hardware on which the software will be displayed and its affect on the nature of interaction proposed.

2.6.1 HCI Principles

Principles that govern good human-computer interaction are an essential consideration when designing the interface of any computer system. There has been considerable research into this area of HCI and as a result a number of principles, heuristics, models and processes have been developed for application when designing and evaluating user interfaces.

The user interface should aim to present information clearly as it acts as the locus for interacting with the system. As Preece et al (2002) state "…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 if the system. This is particularly salient with child users as they must find the system highly usable if they are to benefit from its educational content.
The key considerations when designing a user interface that are relevant to the tutoring system are given below, with reference to Dix et al (2000) and Preece et al (2002).

Use of Colour

Colour can be used to distinguish between objects and also to denote meaning in an interface, for example red is often associated with danger or warning. Importantly it must be used consistently so to represent the same meaning in all contexts, otherwise the user may become confused or misled. This is especially prominent for any visual feedback or animation that the tutoring system provides.
Colour also has a clear relation to perceptual salience such that it can be used to draw the attention of the user to important aspects of the interface for interaction purposes. Under these circumstances figure-ground[4] perception is an important consideration, such that the distinction between figure and ground is maximally represented and is therefore readily interpretable (for example black on white).
We must note that the use of colour is an issue in interface design as its over use can be distracting and also contributory to visual fatigue (Preece et al, 2002). Moreover, the physiology of the human eye makes it difficult to focus on particular colour pairings, for example red and blue are known to be difficult for people to focus on.

Use of Sound

The use of sound can add value to an interface as it forms an expressive means by which to convey the system state or provide feedback from completing a task (Dix et al, 2000). It is advantageous as it can be used to indicate that an action has been performed for example, thus irradiating any user uncertainly. However there is a need to consider that all sounds need to be distinguishable so that users can understand what they represent. Sound should also be used sparingly so not to cause annoyance.

[4] 'Figure' denotes the items of the interface that we *look at*. 'Ground' represents the other aspects, including the background.

Recorded speech can be used to supplement or replace visual information, which is beneficial for foundation stage children due to the natural human prosody and pronunciation. As we have discussed, speech is more accessible to children at the foundation stage because they cannot easily decode textual information. Therefore it is their primary means for interpretation. Using speech to enable communication through the interface also enables us to mirror the natural teacher-child interaction style of the classroom. However, the same considerations of overuse and audibility need to be considered.

Use of Text

Using text to display information is widely used in user interfaces due to its permanence and potential unambiguity. Dix et al (2000) state that text should be used considerately in an interface to convey meaning where it is clearly legible and distinguishable from the interface background. Furthermore, over-use of type faces should be avoided. With child users it is clearly not appropriate to use text to dictate instructions at length; however it can be used to highlight key words and thus add emphasis to the key point of the task.

Required Use of Memory

George Miller's theory of remembering (1956) is important to interface design. He stated that the human mind can only remember 7 'pieces' of information, plus or minus 2 pieces of information, due to the mind's tenancy to 'chunk' information on intake. This has the following implications:

- Not to overload users with complicated procedures for carrying out tasks. This is especially prominent with child users where intuitive task processes are necessary due to their low computer interaction experience.
- Design interfaces that promote recognition rather than recall of information and task processes. Miller stats that colour, icons, graphics and consistently placed objects can promote recognition and thus ease of use.
- Dynamically link representations and abstractions that need to be learned. This is important for the tutoring system, as it has been shown that linking abstract concepts to concrete representations aids children's ability to access and understand that concept more precisely. Cues such as icons and iconic buttons can be used in this context to help 'trigger' information in the user's memory.

Further to the HCI principles we have discussed, we must also consider the structure of the interface and the system's propensity to support the user's tasks and objectives. Clearly poor organisation of tasks and their related processes can result in a user not knowing 'what to do' or in the worst case it can result in them 'getting lost' altogether. Therefore we need to consider heuristics that define *usable* systems.

2.6.2 Principles to Support Usability

Design heuristics can (in theory) be directly applied to the design of the user interface in order to produce a highly usable system. With reference to Dix et al (2000) we can recognise the most important and relevant heuristics - refer to Literature Review Appendix, section 1.

As stated by (Mullet et al, 1995) "Interfaces should be designed to be simple, perceptually salient and elegant, so to adhere to usability design principles". Therefore it can be reasoned that simply applying the design heuristics stated by Dix et al (2000) *may* result in a *usable* system. This can be considered the case on some level, however these heuristics do not provide details of how we make the interactive nature of the interface more usable or how we integrate the theory we have already discussed over motivation, learning and cognition, and thus further consideration is required.

Nielsen (2001) defines usability as the extent to which a product can be used by the users to achieve their goals with effectiveness, efficiency and satisfaction in the specified context of use. Essentially this means that usability is an attribute of the way in which a particular person and a system interact. In order to support this notion Nielsen has developed ten usability heuristics that are applicable to the design of successful user interfaces. These heuristics add important detail to those given by Dix et al (2000) that are particularly important when designing for child users. The heuristics reported here are those considered relevant to the tutoring system:

- Visibility of system status - The system should provide appropriate feedback, within a reasonable time to keep the children and teachers informed about the current state of the system. Feedback should be provided in response to user input to allow them to grasp whether they have completed a task correctly, or to indicate some error. Also intermediate feedback can be equally useful to show which graphic elements have been selected and moved or to indicate that part of the task has been completed successfully for example.
- Match between system and the real world - The system should “speak the users' language”, such that it uses words, phrases and concepts that are familiar to the children and follow real-world conventions. Thus inappropriate mathematical terminology should not be used when dictating instructions or giving feedback on a task. Rather it should be simple and concise allowing the child to follow it easily.
- User control and freedom – Nielsen states that users often choose system functions by mistake and thus perform accidental actions. Therefore there is a need for a clearly marked ‘emergency exit’ allowing the user to leave the unwanted state easily. This can help to relieve the anxiety of users as they know that unwanted states can be exited at any time.
- Consistency and standards - Users should know that the same action, words or situations will always have the same meaning. This invokes confidence in the children and encourages self-directed exploration of the system.
- Error prevention – It is important to carefully design a system with the aim of preventing problems from occurring, or reducing their likelihood. This is especially pertinent with child users who could be confused by error messages that require corrective action.
- Recognition rather than recall – As an application of research by George Miller (1956), the system should promote recognition rather than recall as it is easier on the user in terms of cognitive load. Children should not be required to memorise actions or information from one part of the dialogue to another and thus all relevant information needed at any given point should be visible.
- Aesthetic and minimalist design - The user interface should be simplified as much as possible. By removing unnecessary features, control and information, we reduce the potential for the user to misunderstand the system and become confused. This is important for child users as there is a need for them to be able to concentrate upon the learning task and receive feedback without being disturbed by other system features.

A further consideration for task-based systems is the need to provide ‘closure’. Dix et al (2000) state that sequences of actions should be organised into groups such that the user can gain satisfaction from accomplishing the task set. Subsequently the system should provide some indication that the way is clear to prepare for the next group of activities or tasks. Closure has a clear relation to the tutoring system as there is a need for children to receive feedback on a lesson so that they can adjudge their performance before moving onto the next lesson. Where the lesson has not been successfully completed it may be necessary to repeat it, or alternatively it may be appropriate for the child to expand their knowledge in tasks of increased difficulty. As Dix et al describe, providing points of closure such as these, allows the user to clear their mind and thus reduces the load on memory.

2.6.3 Interaction Styles

The interaction between the user and the computer can be seen as a dialogue of communication. The style of this interaction has an enormous affect on the dialogue itself, as well as the usability of the system as a whole. 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

The user interface is operated mainly with the aid of function keys or the keyboard. Examples of key modal styles are:

- Menu driven systems, where the user interface displays a set of options and the user selects an option using a mouse, or numeric or alphanumeric keys. The selection of an option may generate a further set of options.
- Question/Answer and query dialogue, where the system presents a series of questions in text form, and the user enters the answers via a keyboard.
- Function-key interaction, where the user makes a series of inputs using function keys, prompted with information displayed on screen.
- Voice-based interaction, often supported by a telephone, where the user is presented with options by a recorded voice message. The user makes choices using the telephone key pad, in response to voice instruction.

Linguistic

The style of interaction dictates that all of the user's inputs are made using the keyboard, or a set of conventions such as natural language. Examples of linguistic styles are:

- Command-line interfaces, which allow the user to express instructions to the computer directly and receive feedback in text format.
- Natural language, where the user enters information by speaking words or phrases directly at the computer, receiving feedback in the same language. This mode of interaction is still advancing and is complicated due to the ambiguity of the English language.

Graphical Direct Manipulation

The user interface displays a number of objects on the screen, and with the use of a pointing device (the mouse for example) the user can apply actions directly to the objects of interest. This allows the user to manipulate objects as desired, with few constraints. Examples of direct-manipulation styles are:

- Form-fills and spreadsheets. The user is presented with a set of text fields, allowing the user to select individual fields and enter or modify their contents. The screen essential presents a 'form' resembling its paper equivalent.
- Graphical direct manipulation, which has two closely related interactions styles:
  - WIMP (windows, icons, menus and pointers) is the most common form of interaction where the user directly manipulates on-screen objects through the use of input devices, as demonstrated in Windows operating systems.
  - Point-and-click interfaces. Users navigate the interface via point-and-click operations on hypertext links and iconic buttons, using the input devices. This is the mode of interaction for the World Wide Web.

Given that young children are the target users for the tutoring system it is clear that the interaction style must suit their level of computer literacy. Therefore it would be impossible to use a linguistic interaction style as both command-line and natural language styles require the user to have a detailed knowledge of commands and actions that the system can interpret and respond to. Also given the nature of the task it would be impossible to use a key modal interaction style because menu driven, question/answer, function-key and voice based styles would not support the multimedia framework desired.

Graphical direct manipulation is 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. Shneiderman (1998) describes the features of direct manipulation as:

- Visibility of the objects of interest
- Incremental action at the interface with rapid feedback
- Reversibility of all actions, to allow exploration without penalty
- Syntactic correctness of all actions, so that every user action is a legal operation
- Replacement of complex command languages with actions to manipulate directly the visible objects.

Shneiderman's explanation has clear relations to the required interaction style for foundation stage children. It allows us to recognise that direct manipulation of *visible* objects in the task domain provides the means for directly translating a child's thoughts into computer-based input that can be understood. This helps to bridge the gap between the children's thought processes and the system state[5], which is of clear importance with novice computer users. Also, where rapid feedback is received from performing actions on objects (the objects move around on the screen for example) children can learn to exert control over the task, which Newton (2001) promotes; "learners are more likely to engage in, and benefit from a task if they feel they have control over it." Therefore we can recognise that highly visual tasks and direct manipulation are beneficial to the tutoring system.

Under close inspection we can recognise that features of the WIMP and point and click interaction styles are particularly relevant. The majority of multimedia systems use point-and-click interaction styles because it allows most actions in the interface to be operated with a single click of the mouse button. 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". For example when the user wants to exit the system they might click on an open door, or to delete an item they might put it in the rubbish bin. Clearly the aim of this style of interaction is to reduce the learning time for interacting with the computer so that the user can make inferences about the action performed when they click a button or a hyper-text link. 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.

Point-and-click interfaces are very closely related to WIMP interfaces. WIMP interfaces have become a very common interactive environment for computer systems, especially with desktop Personal Computers because of the emergence of Microsoft Windows and Mac OS. In these operating systems WIMP provides a graphics intensive environment to help the user access the functionality of the system without having an in depth knowledge of its architecture. Clearly windows and menus of WIMP interfaces are not relevant to the task domain as they require the user to understand issues of multitasking and also to have a firm understanding of hierarchical menus and the actions performed by clicking menu items. This is well above the schema of children in the early years.

However, 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. Pointers are also an important aspect of the WIMP interaction. Pointing, selecting and manipulating objects on the

[5] Norman (1998) describes the mapping between user thought processes or intentions, and the system state as cognitive engineering. He states that "the user must translate goals perceived in psychological terms to actions suitable for the system", and thus bridging this gap makes a system more usable and more capable of naturally supporting the user's goals.

screen by use of the mouse as an input device, provides the most natural form of human interaction with a computer because actions are fed back immediately. This is reinforced by Shneiderman (1998), who promotes the development of interfaces, where “each action produces a comprehensible result in the task domain that is visible in the interface immediately.”

The interaction style must aim to make the dialogue between the child and the computer as natural as possible, so to reduce the cognitive load and confusion for the child. We have already discussed methods to do this through themed interfaces and real-world contexts for the application of mathematical skills, but this must be mirrored in the interaction style if the system is to be truly usable. An appropriate combination of the aspects of point and click, as well as some aspects of WIMP interfaces should allow children to perform tasks in the ‘obvious’ way so that they can concentrate on applying their skills and not *using* the computer.

2.6.4 The Science of HCI

Psychological research in the HCI domain has provided some notable advancements in understanding human behaviour in consideration of computers and interfaces (Landauer et al, 1997). We can relate this research to improving system usability in the tutor.

The Hick-Hyman law is an important consideration when designing any computer system, but can be considered especially prominent when we are designing for child users. It dictates that user decision time is proportional to the log of the number of equal alternatives, implying that at the interface level, a single screen with many choices is better than a series of screens with few choices each (Landauer et al, 1997). Therefore we can recognise that the tutoring system should provide the child with a clear and concise set of options for using the system in a single screen, so to aid the transparency of the system’s functionality and to reduce the cognitive load on them.

The computer related skill-base of child users is a further consideration when designing the interface for the tutoring system. Fitts’ Law (1954) is highly relevant in this context as it is used to predict the time it takes to reach a target of an interface using a pointing device. This takes account of human factors research to adjudge the relationship between speed and accuracy when moving towards a target on a display, in consideration of the time it takes to point at a target, based on the size and distance of that object (Preece et al, 2002). Therefore it can help us decide where to locate interface buttons, what size they should be and how close together they should be on a display.

Essentially we can see that bigger targets can be accessed faster and with less need for accuracy. Clearly this is of key importance for child users whose control over input devices will not be as well developed as experienced computer users. Hence interfaces with large, well-spaced buttons will be vastly more accessible for the children than interfaces that present a lot of small buttons in close proximity. We can also consider using buffer areas so that child users have less opportunity to hit another button if they accidentally go past the target. These considerations will reduce the propensity of the interface to frustrate the child where accidental actions are performed.

2.6.5 Anthropomorphism[6] and Animated Agents

A general assumption made by designers of software for novice users is that they feel more at ease with an interface ‘persona’ (or companion) who provides instruction and encouragement. It is considered that personas can convey emotional states and elicit certain responses from users, which can provide confidence and help to reduce the initial learning effort a user has to overcome.

[6] The propensity that people have to attribute human qualities to objects.

Preece et, al (2002) support the notion that interface personas make human-computer interaction more enjoyable for novice and child users. Furthermore Preece et al state that interface personas can help to enhance meaning, and invoke motivational responses in children. Motivating the child to learn is a primary concern for the tutor, which we have already seen can be achieved through the use of an appropriate interaction style, an intuitive interface and use of real-world contexts and scenarios for applying their skills. However, each of these factors is determinant on the child understanding what is expected of them. An interface persona could provide instruction and support as a 'tutor' within the tutoring system, and thus could give feedback and encouragement in relation to the performance of the child.

"Working together can increase [children's] motivation, effort and persistence; working together can also increase their confidence in their own ability to succeed" (Segers et al, 2002). As previously stated, an interface persona could enable collaborative learning within the tutoring system. By using mathematical language in appropriate situations through speech and text, the persona could help the child to gain a deeper understanding of the correct context for its use. Also by supporting the child throughout the tutorial, the persona can relay feedback of whether they answer questions correctly and provide the correct answer if they are wrong. Thus a persona could be used to direct and manage the task set by providing a focus of interest for the child as well as providing feedback and instruction through the interface.

There is a need to consider that anthropomorphic agents and interface personas are a contentious issue in interface design and HCI. Shneiderman (1998) argues that interface personas can make users feel inferior if their mode of instruction is too patronising. He states that users can become disillusioned with the software tool if the persona inhibits interaction and annoys the user. Therefore the persona 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. In the supportive and instructive roles already stated, the persona provides clear benefits because it can effectively convey emotional states. This may help the child to more clearly understand spoken instructions as well as the feedback they receive. However as Shneiderman states, it must not intrude on the system in such a way that it inhibits the child's natural thought processes, or ability to answer a question unaided.

2.7 Conclusion

Through research conducted into several areas relevant to the problem domain, we have raised many interesting issues that have formed a theoretical base to the project. There is a need to investigate many of these issues in more depth to assess whether they are prevalent in the foundation stage classroom. However we can begin to highlight those that are relevant to designing computer-based educational software and the extra deliberation that this may represent.

Firstly, we must consider the role of the tutoring system. As we have discussed, computer-based tutoring systems are well suited to supporting educational development as they can effectively provide unlimited opportunities for children to practice applying their mathematical skill base. Importantly, the aim of this tuition must be to enact the benefits of accelerating development (Bruner, 1967) so that children's understanding of important mathematical concepts improves, while consolidating activities of the classroom. We can recognise that this role is not mutually exclusive from that of the teacher. Teachers clearly are responsible for introducing new concepts under the guidance of the national curriculum and thus the tutoring system should provide the application for practicing these concepts.

More specifically the suitability of computer-based multimedia for fulfilling an educationally supportive role is that it provides the basis for the manipulation and exchange of information in a fundamentally different form to that of the classroom. Multimedia software can provide a visual perspective on abstract concepts, allowing child users to rely on their sense of visual perception, which both Piaget and Liebeck (1984) describe as being particularly prevalent in foundation stage children.

Of course, there should be no presumption that the use of the computer by its nature will offer a better tool for learning or make information more readily interpretable. If the system is to be truly usable in the classroom there is a fundamental need for the learning material that it poses to be "…relevant to other situations for which the child has well developed schemas" Copeland (1979). This point was reinforced by Vygotsky and Donaldson (1978) who state that there is a need to link new abstractions to previous experiences and real-world contexts to make them accessible for children. Therefore there is an inherent necessity to investigate learning metaphors that children would consider appropriate and stimulating, that could subsequently be applied in the tutoring system to aid children's understanding and make the system more accessible for applying their knowledge. Accordingly, it is also necessary to ensure that we gain a good understanding of the national curriculum and the mathematics objectives that it sets out, so that the system provides appropriate educational content.

When considering of the mathematical content that the tutoring system provides there is a need to discover the best method for integrating the system into the lessons structured by foundation stage teachers. The national curriculum is important here, but there is also a need to interview teachers and observe classroom activities so that we can assess the best methods for delivering and displaying information to the children in ways that they are used to and can naturally interpret, with the aim of making the system both highly usable and educationally valuable.

With reference to Keller (1983), effective learning sessions are enacted through stimulating the learner and motivating them to complete tasks. Keller states that this is borne out of the ability to gain children's attention, provide relevant material, inspire confidence and allow the user to gain satisfaction from completing tasks and getting feedback and rewards for positive results. By using appropriate mathematics material and learning metaphors we have already identified methods by which to provide relevant material and gain the child's attention. However, tasks must also be incongruent when performed time and gain so that their learning potential does not deplete and appear static to the children, so that it becomes more of memory task than an educational one. Moreover appropriate feedback and a reward scheme were identified as necessary to inspire confidence and gain satisfaction from allowing the user to achieve their goals.

We have discussed the relevance of HCI principles and usability heuristics to making the system intuitive in consideration of children's computer literacy and interpretive abilities. This knowledge should be applied throughout the design stage 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. We should also apply these principles and heuristics when evaluating current tutoring systems both to see if they are compliant and also to identify their well-designed features. By interacting with these systems we can avoid simply reinventing what is already available, and thus reiterating their short-comings. Moreover we can build upon and improve their well-designed features and apply these in the design and implementation of the tutoring system.

There is a need to consider how the software will support the computing skill base of foundation stage children, which is an important area for further investigation. Being the first year of school the children may not have a vast breadth of experience with computing hardware and thus their skills for operating input devices and interpreting information through the display will not be as well developed as that of older children. This may affect the available mode for interaction, which we have already identified as needing to be graphical direct manipulation so that all user actions are fed back quickly and clearly. Accordingly this may impact the mode for user input which may be restricted to simple mouse and keyboard actions for example.

As stated at many points in the preceding discussion, empirical studies and investigation into relevant practical sources must form the 'next stage' so that we can examine the common practices of teaching and learning processes in its natural environment. Information gathered from these sources when coupled with the theoretical base established in this chapter will allow us to recognise the most important problems that need to be addressed in the design stage in order to produce the required highly interactive mathematics support tool.

3 Requirements Analysis and Specification

3.1 Introduction

The literature review has enabled us to begin thinking about the issues that are important to the design of a computer-based tutor. It has also uncovered valuable research detailing the methods that best support learning and teaching processes at the early year's level and thus provides the justification for using a computer-based tutor in an educational context. However it is important to balance this theoretical base with an understanding of more practical issues surrounding system stakeholders and existing tutoring systems evaluation.

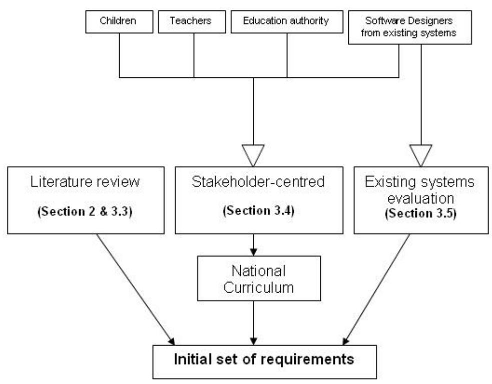
By undertaking empirical studies we are able to identify the needs, expectations and goals of the stakeholders of the final system such that we can appreciate the structure of the activities they undertake and the rationales for performing tasks in certain ways. Furthermore we can gain an insight into the conditions under which the product will be used and the constraints on its performance, with a view to maximising the potential of computers and multimedia software as tools for learning.

The requirements analysis process enables us to produce a "stable set of requirements that form a sound basis to move forward into thinking about design" (Preece et al, 2002). Moreover, the emergence of a detailed set of requirements provides a framework for effective evaluation of the final system as a solution to the problem space.

3.2 Sources of Requirements

Generation of system requirements by its nature must involve an in-depth analysis of the problem domain. As shown in figure 3.1, it is necessary to involve a wide spectrum of people and other sources in this analysis so that we establish a complete set of requirements from theoretical and practical backgrounds that can subsequently be taken forward to the design stage.

Figure 3.1 Diagrammatic representation of the sources of the system requirements



The requirements sourced from the literature review are detailed in section 3.3.
As already stated, the literature review has provided the theoretical base that drives further research into the practical issues related to the problem domain. As well as specifying a set of requirements gathered from this research activity, we can apply the salient points throughout the

requirements analysis process to elicit further requirements from stakeholder and existing tutoring system sources.

Stakeholder-centred requirements analysis must involve all parties that impact the problem domain. It is an important activity as there is an inherent need to understand the goals and activities of the stakeholders so that we can effectively support their tasks and expectations. This is especially important with child users as we must aim to support their learning needs in consideration of their computing and mathematical skills. Thus we must aim to ensure that children do not have to change the way they perform tasks or have the need to learn new skills to suit the processes of the tutoring system. We must also consider the needs of teachers and their lesson planning so that we can produce a tutoring system that is capable of being integrated into mathematics education in the early year's classroom.

A key part of the stakeholder-centred requirements analysis is understanding the role of the educational authority in governing early years teaching media. Together with the view of industry professionals (software designers for existing tutoring systems) this allows us to gain an understanding of important features that a computer-based learning tool must provide as well as key problems and constraints on its performance. Moreover we can exploit the knowledge and experience from these sources throughout the design stage.

Stakeholder-centred requirements analysis will involve gathering data from the following sources:

- Children at the foundation stage at a representative infant school (i.e. North Baddesley Infant School, Southampton). In empirical terms this encompasses any child around the age of 5, and certainly in the range 4-6 years, where the foundation stage represents their first year at school.
- Teachers for the foundation stage at a representative infant school (i.e. North Baddesley Infant School, Southampton).
    - Louise Aylott, a recently qualified teacher who has graduated from a teacher training degree and has 2 years experience of teaching foundation stage children. Louise can provide an insight into how she has applied the theory base from her degree in a practical teaching role at the foundation stage.
    - Janis Reeves, an experienced teacher of 23 years who has taught at both the key stage 1 and key stage 2 levels, encompassing children in the age range 4-8 years. Janis can provide a wealth of knowledge of the methods she considers best to teach children at the foundation stage. She can also provide a detailed knowledge of computer-based teaching media having used it in the classroom for many years.
- The education authority:
    - Pat Lamb, an LEA advisor who assesses the quality of foundation stage teaching in relation to the national curriculum for Hampshire schools. Pat also has vast experience of teaching as her previous job was as a head-teacher for primary age children. Pat can provide an insight into the relation between governmental requirements for computer-based teaching media and the best way to exploit multimedia in a learning context. Through her experience as a teacher, Pat also offered valuable advice on the required role of a computer-based tutor in lesson planning at the foundation stage.
    - Joyce Chester, a recently retired ICT coordinator for Hampshire schools, with focus on mathematical tooling. Joyce can provide details of tutoring systems that are currently used in Hampshire schools as well as advice on aspects of tutoring systems that are valuable educationally, or that children find particularly engaging.
- Educational software designers, for packages used in North Baddesley Infant School for mathematics tutoring:

- Jan Burd – 'My First Maths Adventure - Counting and sorting' – DK Multimedia. Jan is a software designer who was part of the team who developed 'counting and sorting'.
- Sue Rankin – 'A Teddy Bear's Picnic' and 'Tizzy's Toybox' – Sherston Software. Sue was the lead software designer for both these projects and has a vast amount of past experience from working with teachers and software development teams on a number of tuition solutions.
- Rod Boyes – 'Millie's Maths House' – Inclusive. Roy works as a procurer of educational software for Inclusive, and is a member of the team who develop updates and translations for software imported from America.

The methods applied to gather the required data are detailed in section 3.4, together with the set of requirements elicited from this activity.

As shown in figure 3.1 a further activity that forms part of the requirements analysis is the evaluation of existing tutoring systems. The essential details of the evaluation process, using the appropriate design and usability principles and heuristics are detailed in section 3.5, together with the requirements gathered from this activity.

The sources identified for the requirements analysis provide a breadth of experience and knowledge, from which we can elicit the system requirements. However throughout this process there is a need to consider Carroll's (1991) task-artefact cycle. Carroll stated that empirical studies allow us to scope the tasks that users *currently* perform and those they *want* to perform. He stated that when this knowledge is translated into a new system further observation can reveal that users require support in additional activities, or their original needs change. This can enforce the generation of a new set of task definitions that act as the requirements for future or updated systems. Of course this life-cycle is natural for any computer system, but its relevance is clear to the computer-based tutor as we can acknowledge that the initial set of requirements may change over time, thus enforcing the need to involve the user throughout the design and implementation processes.

Both to support the task-artefact cycle and to ensure that the requirements gathered from all sources represent the key issues and goals that need to be supported by the final system, it is important that the requirements are checked and validated by the user. This validation process was performed with Louise Aylott and is detailed in section 3.6.

3.2.1 Specifying the requirements

The framework we will apply for structuring the set of requirements is taken from Preece et al (2002). Traditionally requirements can be thought of as either functional (capturing what the product should do) or non-functional (the constraints on the system and its development), which provides the basis for grouping like requirements. However Preece et al state that these categories are too broad when specifying requirements for interactive systems and so they offer a more complete set of categories that allow us to group requirements in a more appropriate manner.

The categories appropriate to the tutoring system are given below. These will be applied to structure the requirements gathered from each requirements elicitation activity.

- Functional requirements: capturing the functionality that the system should provide.
- Non-functional requirements: capturing the constraints on the system's performance and its functionality, as well as facts and assumptions relevant to the system.
- Usability and user interface requirements: capturing the important usability goals and associated measures for the system.

- User Requirements: capturing the characteristics, abilities and skills of the intended user group.

3.3 Requirements gathered from Literature Review

This section details the requirements gathered from the literature review (section 2).

Functional requirements

1. The system should display a complete set of the available task options and customisability options for the user.
   Description: The user should be presented with a complete and concise set of options for using the system (tasks and activities) in a single screen, to reduce the cognitive load on them, as inferred by the Hick-Hyman law.

2. The system should present tasks using appropriate learning metaphors.
   Description: Tasks should use appropriate learning metaphors so that they are readily accessible for child users. Learning metaphors should help to bridge the gap between the abstract mathematical concepts and concrete applications.

3. The system should provide a set of tasks that incrementally increase in difficulty so to support the user's educational development.
   Description: The user should be able to apply their knowledge in more difficult scenarios so to nurture and expand their level of understanding of the mathematical concept at hand.

4. The system should use graphics and animation to enable the presentation of highly visual tasks for the children to apply their knowledge within.
   Description: Tasks should promote the use of visual perception for interpretive purposes, due to the sensorial dominance it has over foundation stage children. Also graphics intensive tasks should by their nature be engaging and fun for child users, and thus help to maintain an efficient learning session.

5. The system should use an interface persona to direct and manage the task set.
   Description: A persona should be used as the source for communicating audible instructions and feedback to enact the benefit of collaborative learning. The persona should not be intrusive to the functionality of the system or inhibit the child's natural thought processes.
   In relation to Nielsen's usability principle stating the need for a suitable match between system and the real world, the interface persona should "speak the user's language", enabling child users to practice listening to and subsequently using mathematics vocabulary in appropriate contexts.

6. The system's mode of communication with the user should have a 'permanent' nature.
   Description: Further to the audible communication given by the interface persona, key words should be printed to screen via text to present the salient points of the instruction or feedback. This should act as a reference for the user on the aim of the task or the main message of the feedback.

7. The system should provide appropriate feedback to the user.
   Description: This relates to Dix et al's design principle of system responsiveness. The tutor should provide feedback to denote whether the user has answered a question rightly or wrongly, allowing them to gauge the level of their achievement.
   If a correct answer is given, feedback should be positive and encouraging so to enhance the user's satisfaction from completing the task. As Keller (1983) states feedback of

this nature positively affects a user's level of motivation. If a wrong answer is given, the correct answer should be revealed, providing the basis for the user to recognise and possibly reform their mistake.

8. The system should allow the user to control all tasks.
   Description: In relation to Nielsen's usability principle of user control and freedom, the system should have an 'emergency exit' that allows the user to leave a task at any time.

9. The system should allow the user freedom to self direct the learning session
   Description: As promoted by the DFEE (1999), users should have the freedom to explore the interface and its tasks, thus enabling the benefits of ICT media in a learning context. Moreover the user should be able to specify if they want to apply their knowledge base in more difficult tasks (relating to requirement 3).

Non-functional requirements

10. The system should be an effective teaching tool allowing children to practice applying mathematical concepts taught in the foundation stage classroom.
    Description: The mathematical content of the tasks presented by the system should be clearly mapped to the foundation stage national curriculum as well as classroom lessons so that concepts are not introduced above the child's schema.

11. The system should be reliable and not produce any unexpected behaviour.
    Description: The interface should comply with Nielsen's usability principle of error prevention. Thus comparable actions should have predictable results so to avoid unexpected system behaviour and user surprise. Any exceptions generated by the system should be handled in a way such that the system does not fail.

Usability and user interface requirements

12. The system should instantiate a graphical direct manipulation interaction style.
    Description: An appropriate combination of point-and-click and WIMP interaction styles should be used to promote a high level of user interaction in an intuitive interactive environment. This is the most appropriate interaction style given the computer literacy, the computer-interaction experience of foundation children and the nature of the problem domain.

13. The interface should allow direct manipulation of objects for relevant tasks, so to provide a different perspective on how to arrive at an answer in a mathematics context.
    Description: Direct manipulation of objects provides the means for highly interactive tasks where the user constantly inputs instructions and receives rapid feedback on the status of those instructions. Where the user can apply actions directly to objects in a highly responsive interface they can also learn to exert more control over them, which Newton (2001) states as important if a system is to engage a child in a learning session.
    Direct manipulation of objects also supports the previous requirement of providing visually-rich tasks (requirement 4).

14. The user interface should be easy to understand, operate and navigate.
    Description: Users should be able to apply 'obvious' actions on objects such that their natural thought processes are easily translated into input at the interface. Thus the interface should promote recognition of how to perform actions rather than requiring the user to recall information from memory.
    The system's navigational structure should be predicable and so that navigational actions do not cause surprise to the user. Accordingly principles governing usability, HCI and design should be applied and adhered to so to create a highly usable interface that elicits positive responses from the user.

15. The interface should be highly learnable, allowing users to easily pick up how to use the system.
    Description: Colour, sound and text should be used consistently to represent the same meaning throughout the interface so to prevent user confusion. These media should enable efficient user interpretation of instructions and feedback as well as aid the system's overall usability.

16. The system should be accessible to more than one user at a time.
    Description: Children at the foundation stage are encouraged to develop shared understandings with other children and adults, as it is proven that they learn effectively in collaboration. Thus the system should be accessible to multiple users allowing them to discuss tasks.

User requirements

17. The system should support the user's level of mathematics competency.
    Description: The mathematical content of the system should be clearly mapped to the foundation stage national curriculum. It should not introduce new topics or present scenarios that are inappropriate given the mathematical abilities of foundation stage children.

18. The interface should allow a large margin for error to compensate for the user's potentially limited control over the mouse and thus the screen pointer.
    Description: The interface should have big, well-spaced buttons and buffer areas to prevent users from launching accidental actions, as stated by Fitt's Law.

3.4 Stakeholder-Centred Requirements Analysis

This section details the data collection and analysis techniques that have been used, involving system stakeholders, to aid requirements elicitation and direct thought about the design of a computer-based tutoring system.

As we have discussed in the Sources of Requirements (section 3.2), the stakeholder-centred requirements originate from four sources, the children, the teachers of the early year's curriculum, the education authority for early years and the software designers for existing tutoring systems used in the classroom. To elicit requirements from these sources three complementary data gathering techniques were combined, with reference to Preece et al (2002):

Observation

Naturalistic observation is necessary because it can be difficult for users to explain how they go about achieving their tasks when they are not performing them. This is especially true with children, who may not be able to articulate how they use the computer per se. Thus naturalistic observation allows us to gain a fuller understanding of their activities as they happen, in their natural environment (the classroom). Naturalistic observation is also appropriate for gaining an insight into mathematics teaching in its natural context, where the teacher is not inhibited or pressurised by recording equipment.
The observation sessions used were:

- Children using computers: allowing us to understand their skill set regarding computer hardware as well as aspects of computer-based media that they find engaging and enjoyable.
- Classroom lessons: allowing us to understand mathematics teaching methods that are accessible to children, as well as methods that teachers use to gain the attention of children in a learning session.

Interviews

Interviews enable us to explore relevant issues in depth, given a set of prepared questions that enable us to direct the discussion. Through discussion we are able to elicit scenarios of use for the system, as well as potential problems that we must consider in the design stage. The interviews undertaken concerned:

- Teachers: aimed at understanding the structure and content of lessons, and the proposed educational benefits of particular methods of teaching. Also it is necessary to gauge the required role of a computer-based tutoring system as a tool to support learning.
- The education authority: aimed at understanding governmental requirements for a tutoring system and aspects of tutoring systems they consider important given the requirements of the national curriculum.
- Software designers: allowing us to exploit their knowledge and experience concerning software development in an educational context

Studying documentation

Studying documentation enables us to understand the rules and procedures related to the task, without the need to involve and possibly disturb system stakeholders. There was a need to study:

- The national curriculum: allowing us to understand the background information and regulations governing mathematics teaching and ICT at the early year's level.

The following sections detail the knowledge gained and requirements elicited from collecting data from the system stakeholders. The knowledge gained from studying the national curriculum is applied throughout, where it is particularly relevant.

3.4.1 School Visits and observations

Although the aim is to provide support for teaching activities, we must be aware of the goal of early year's teaching as a whole; to develop the knowledge of children. Therefore we can recognise that a computer-based tutoring system has a part in enabling this goal to be achieved and thus there is a need to understand how learning is best supported, by observing practical lessons and teaching in its natural environment.

Classroom Activities and Lessons

The most important lessons of the school day are core teaching activities led by the teacher. During this period the teacher teaches small groups of children and monitors their progress against the objectives of the national curriculum. This period forms the basis of direct teaching, where new concepts are introduced, misconceptions are reformed and feedback and encouragement are given to the children. Other activities conducted in the classroom are mainly based around practicing the skills developed in this time.

A core activity I chose to observe was a mathematics lesson called "number in the bag". This lesson was a good example of a learning session where the teacher tested the children's knowledge of the number sequence whilst reforming their mistakes and assessing their understanding. The lesson had a clear mathematical mapping to the national curriculum, such that it aimed to develop children's knowledge of number, the number sequence, addition and subtraction.

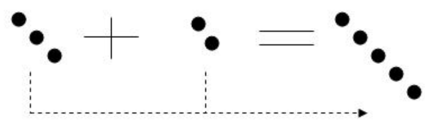
The lesson enabled groups of children to practice simple addition and subtraction operations on the number sequence 0 to 10. The children each had a fan of numbers (0 to 10) and when the teacher picked a number out of a separate bag they had to find the matching number on their fan. The teacher then increased the difficulty of the task by asking the children to find a number on their fans "one more" or "one less" than the number, and later, "two more" or "two less". Important observations from this lesson were:

- The instructions given by the teacher to the children were very concise and simple which aided their understanding. The teacher also gave a lot of feedback on what the children were doing in the early stages of the lesson to provide encouragement and confidence.
- Using a fan as a learning metaphor was beneficial for the children as they had a physical entity to manipulate when arriving at an answer.
- The children got better at the task as time went on and their understanding developed. This enabled the teacher to increment the level of difficulty of the task as previously described. We can relate this method of teaching to the tutoring system as there is a need for the tutorial to increment in difficulty so that we nurture the child's increased understanding and provide a challenge.
- The children benefited from being in a group situation. Some children when unsure of the answer to a question discussed how they might arrive at answer with another child, or copied others in the group to correct their mistake. As we have discussed in the literature review, collaborative learning is an efficient learning method that children enjoy. We have already stated that this relates to the tutoring system through using an interface persona such that children can respond to its instructions and copy its actions as a paradigm for learning.
- The children were encouraged by the teacher to use the correct mathematical language when they had decided on an answer, for example "3 is 2 less than 5". This consolidates their understanding and enables them to articulate their thought processes correctly. It also allows the teacher to verify that the children understand the mathematical abstraction behind the activity.

A second core teaching I chose to observe was a lesson to introduce addition to the class. This was a valuable observation session as it demonstrated the methods used to introduce a new mathematical concept, namely addition. The teacher used a learning metaphor as the basis for teaching this lesson, in order to make the concept more accessible for the children.

The lesson was given at the start of the second term as it builds upon lessons of the first term, including number recognition and the number sequence. The lesson involved the teacher teaching small groups of children in view of a traditional white board[7] where she was able to draw sums and mathematical symbols. Notably use of language was important in this lesson. The teacher introduced "+" as meaning "plus", where the child must sum the numbers on both sides of the "+" sign. The teacher also emphasised the meaning of the sum as a whole, such that "3 plus 2 is 5" for example. In order for the children to conceptualise this sum in practical terms, they were given a set of small animal figures, enabling them to see that "3 animals plus 2 animals is 5 animals". The children found this easier to comprehend by moving the animals in the sum to the other side of the equals sign, allowing them to count how many were in the "total set".

Figure 3.2 Children moved the items "in the sum" to the other side of the equals sign, and then counted how many there were.



As discussed in the literature review, metaphors for learning are very important at the early year's level so that children can relate abstract mathematical concepts to concrete scenarios in a manner that they find accessible. Using the animal figures to allow the children to manipulate

[7] A traditional white board allows the teacher to draw on a large re-writable surface in view of a group of children.

the sum in physical terms meant that they could more easily to relate the fact that they had to find the *total* of the sum, which clearly aided their understanding. This also helped them to understand the meaning of the mathematical language (i.e. "plus" means finding the total). We can relate this finding to the tutoring system as it should enable the children to directly manipulate objects so that they can more easily relate to the sum. This should help them to easily conceptualise the route to the answer, without necessarily needing to understand the mathematical abstraction behind it.
Later the teacher was able to introduce the use of each child's fingers as a metaphor (similarly to the animal figures) and also use higher numbers to increase the difficulty of the sums.

Songs are an important learning tool for an early year's teacher as they enable to the children to focus their thinking on activities other than the mathematical concepts they represent (i.e. the children focus on the rhythm of the song, instead of the need to add or subtract two numbers for example). Hence, songs are used in a metaphoric context so that all the children can practice applying their knowledge of mathematical concepts in a manner that they find accessible, and without the need for true logical thought. I observed two lessons that involved the children working as a group and singing:

- "10 fat sausages in a pan, one went pop and one went bang". 10 children stand up in the classroom, when "one went pop and one went bang" two children sat down and the children had to recite as a group how many sausages were left (8 in the first instance, then 6, 4, 2, 0)
- "10 green bottles hanging on the wall". 10 children stand up in the classroom, when "one green bottle falls" one child sat down and the children recited how many were left (9 in the first instance, then 8, 7, and so on).

The teacher encouraged children to use their fingers as a prompt for calculating the number of sausages or bottles left in the sequence. This gave them a consistent and justifiable method by which to arrive at an answer, and is commonly applied in many mathematical exercises conducted in the classroom.

The influence of each child's perception was important for this task. Clearly where the child perceives the physicality of a child in the sequence sitting down they can more easily relate to the fact that there is "one less" left in the sequence. As Piaget found through his research, perception is of key importance when educating foundation stage children as it drives their understanding and reinforces their learning. Clearly this can be replicated in the tutoring system using graphics and animation to visualise a problem scenario.

Games form a further metaphoric teaching tool as, similarly to songs, they give the children a accessible focus for the learning session. Through observation of a jigsaw game, where the children had to roll a dice to gather each piece of the jigsaw (if they roll a 5 they get the 5th piece, 4 they get the 4th piece, and so on) it was clear that children enjoy games and have a clear desire to win them. The desire for victory maintains their level of interest and gives them a short-term goal to work towards. However games can also cloud the educational worth of the exercise, as the child's concern over completing the jigsaw was at times more apparent than their interest in completing the task correctly.
It was possible to recognise from this activity that children struggled to count the dots on the small dice and frequently made mistakes due to its small dimensions. This is a clear point for consideration for the tutoring system as all objects must be appropriately sized and clearly defined for the child to find them accessible.
Moreover, it was clear that children had a small span of concentration with game-based exercises as they provided little challenge once they were grasped and understood. Most children became notably less motivated when playing the game for the second time for example, because of the re-use of the same information. Thus we can infer that there is a need for tasks to "change" in terms of their difficulty or style if we are to maintain an efficient learning session with the child. Incrementally increasing the difficulty or changing the style of a task allows us

to provide the required variety and flexibility to consolidate and expand the child's knowledge in different contexts.

More generally all lessons taught were no longer than approximately 20 minutes in length. Through observation and discussion with the teacher it was clear that this period of time represents an efficient learning period, and longer would result in children becoming restless or losing their motivation. Therefore the children frequently rotate classroom activities on a 20 minute basis so that they perform tasks requiring different skills in different contexts.

Computer Usage

Children use computer-based mathematics tutoring packages for practicing the mathematical concepts taught in the classroom, for example counting, number recognition and knowledge of the number sequence. The most frequently used packages in the classroom are Tizzy's Toybox, a Teddy Bear's Picnic, and Millie's Maths House each of which pose questions disguised in various scenarios that require the child to apply their numeracy or literacy skills.

From observing groups of children using these packages it was possible to recognise that:

- There was a noticeable novelty appeal for the children when interacting with the computer, particularly when they answered questions correctly and received positive feedback. The children enjoyed the animations, sounds and colours relevant to certain tasks, which clearly helped to maintain their concentration for the 20 minute learning period. As we have discussed in the literature review, studies performed by the DFEE and Hitcham (2000) have concluded that computers and multimedia are exciting learning tools for children because of their enthusiasm for interaction, which is clearly important if any tutoring system is to be successful.
- All the packages print text to screen as well as using speech output to denote instructions and feedback. However from questioning the children and through observation, it was clear that they primarily relied on speech and sound to interpret the task. Therefore we can surmise that children have the ability to interpret sound and speech, and perceive information displayed on a standard-size computer screen, when communicated in Standard English. Given audible instructions it was also clear that the children were able to perform actions independently at the interface, without teacher guidance or additional instruction.
- The children anticipated feedback instantly on whether they were performing the task correctly and whether they had given the correct answer. Where this was provided it was noticeable that the children gained confidence and were motivated by the local goal of performing a task to achieve positive feedback.
- The children were motivated by reward schemes[8] in the packages and it was noticeable that some children were keen to repeat tasks that they were good at, in order to gain this reward. However some of the packages did not vary significantly enough in terms of difficulty or style when they were repeated, which meant some children became bored and less motivated.
- Some children became frustrated when they could not perform part of a task or the interface did not react in the expected way. It was clear that the children had limited skills to troubleshoot or apply common sense in situations where they could not successfully manipulate on-screen objects as they desired. This relates to the Nielsen's usability principles (2001) such that error prevention should be a primary aim when producing 'usable systems'.
- The children required large interface features and buttons so that there was large margin for error in consideration of their control over the mouse. Some children struggled to manipulate very small on-screen objects which contributed to them becoming frustrated. This relates to Fitts Law (1954) which, as described in the literature review,

[8] A reward scheme is typically an animation that is indicative of the child's achievement. Therefore the more questions answered correctly, the better the animation for example.

essentially dictates that bigger targets in the interface can be accessed faster and with less need for accuracy.

To gain a true understanding of the computer literacy of foundation stage children it is not enough to simply observe them using tutorial packages. To gather a more detailed appreciation of how well children control input devices and react to interface events it was necessary to observe them using the computer in a variety of other situations, in their natural working environment. Therefore I observed them using the painting and word processing packages that are used frequently in the classroom.

The children use a simple painting package that relies on good mouse control to draw teddy bears, fireworks and robots for example, in relation to the classroom theme of the day. From observing the children using this package I noticed they could quite intuitively use the paint box to click on the colour they desired and then paint on a blank document. They could also change the brush being used by selecting a different tool, through a similar point-and-click operation. The level of control over the mouse was generally good enough such that each child could translate their thoughts onto the computer screen via the input device.
Furthermore each child was able to print their work by clicking on the print icon, although it was noticeable that there was no confirmation prompt for this.

A further computer-based activity observed was children using a simple word processor to type their names initially, and subsequently words as prompted by the classroom assistant. The children required assistance when performing this task, as many could not independently operate the keyboard. Through discussion with the classroom assistant it was possible to recognise that the children struggled to decode the meaning of the letters (especially in uppercase), due to their lack of familiarity with them in a computing context, and due to the size of the letters on the keys. From observation it was also clear that the children made errors due to the close spacing of the keys on the keyboard. Therefore we cannot assume that any part of the interaction process can rely on the child being a proficient user of the keyboard as an input device.

3.4.2 Discussions with teachers

Through discussion with Louise Aylott and Janis Reeves and by studying the objectives of the national curriculum, it is clear that mathematics is very important at the early year's level and forms an important part of the period assessments. As previously stated, interviews allow us to explore issues that do not arise through observation and thus we are able to gain an understanding of how teachers deliver mathematics lessons in different scenarios.

Classroom Lessons and the National Curriculum

Through discussion with the teachers it was possible to identify that mathematics is used in a variety of situations throughout the school day. Both teachers emphasised the need for children to become familiar with the basic mathematical concepts such as the number sequence and number recognition, as they form an important base for future teaching. Hence these concepts are practiced in many diverse contexts throughout the school day.

The "job box" is used as the main lesson plan for teaching throughout the school day. Children select an activity from the box from a variety of options, usually including painting, writing, games and computer use for example, each of which has a 20 minute duration. There are also a number of core teaching activities which all children must participate in at some point during the learning session, where new concepts are introduced in accordance with the national curriculum, as we have already discussed (3.3.1 School Visits and Observations). The majority of the activities from the job box have some relevance to the core teaching activity such that a child can practice applying the newly taught skill in a number of different ways and in different contexts.

Each child's work in the classroom must be recorded for the needs of the foundation stage profile[9] and to show each child's parents their academic development and achievements. Computer use and ICT are a key part of this profile. There must be proof that the child has gained and developed their skills, understanding and control over computer software and hardware, and applied this skill base across a range of subjects, including mathematics. Providing evidence of good control over computer hardware is very difficult to feedback through a tutoring system, and is usually demonstrated using pictures the child has developed using the classroom painting package or documents from word processing. However, both teachers expressed a need to be able to access a summary of the tasks and sums that each child completes when using the tutoring system to check that they get answers right (and thus understand how to apply their knowledge), and for the purposes of the foundation profile.

Formal assessments for mathematics are given every half-term (approximately 6 weeks) consisting of individually answered worksheets, one-to-one exercises directed by the teacher and a series of questioning sessions that allow the teacher to ascertain the child's development in accordance with the national curriculum. These assessments consist of sums and practical problems which aim to demonstrate that the child can apply their knowledge and use mathematical language appropriately and in the right context. Thus the computer-based tutor would need to allow the children to practice applying appropriate skills in order to prepare for these assessments.

Role of the Mathematics Tutoring System

Through discussion with the teachers it was established that the preferred role of the computer-based tutor is as an activity from the job box, where the children can practice applying the skills that are taught in the classroom. In accordance with the other job box activities, the tutor should provide a short and well focussed lesson of approximately 20 minutes where *real* educational value is gained while the child is fully motivated and attentive.
Moreover the tutoring system should be able to be used in conjunction with the other computer-based tutors used in the classroom. Louise Aylott in particular stated that the tutoring system should complement the activities presented in these systems.

Both teachers expressed the need for computer-based activities to allow a child to work independently. This means that the tutoring system should provide relevant and engaging tasks that maintain a child's attention so that teacher instruction, assistance or discipline is not required. However, there is also a need for the tutoring system to be accessible to pairs or small groups so that the children can discuss each problem posed and possible answers. The reasoning for this was that there are only 3 computers in the classroom, and thus there is an occasional need for sharing. This relates to Segers et al (2002) who states that group work inspires increased motivation and effort and thus is clearly beneficial in the classroom.

The importance of computer-based tuition for mathematics and other subjects to their lesson planning emerged through further discussion. At key-stage 1 the numeracy hour dictates that the children must spend at least 1 hour per day learning or applying their numeracy skills and thus both teachers clearly valued the opportunity offered by tuition packages to fulfil at least part of this time meaningfully. Being able to allocate small groups of children to using computer-based tutorial packages (or other activities with the classroom assistant) allows the teacher to focus teaching on groups of children based on their age and mathematical ability. This allows them to tailor their methods of teaching more appropriately to the skill level of the group, without confusing some children or being too easy for others. Therefore children use the

[9] An accumulation of work for each child throughout the foundation year.

computer at least once a day, and thus tutorial packages form an important part of their education, as well as teacher lesson plans.

Required Content

It is necessary to identify important mathematical subjects of the national curriculum that require detailed practice or support, which can be offered through a computer-based tutor. Both teachers stressed the importance of closely mapping the tutorial content to the requirements of the foundation stage curriculum, so that concepts are not introduced that have not been taught in the classroom and would consequently be above the child's schema. Thus with reference to the national curriculum for mathematics, we can recognise that children should develop the following skills at the foundation stage:

- Conservation of number and the number sequence
- Mental calculation
- Shape and space
- Introduction to addition
- Introduction to subtraction

As directed by the teachers, number recognition[10] and the number sequence[11] are particularly important lessons of the first term. The teachers stated all children can count to at least 10 (in most cases they can count to 20 and beyond) and have a good understanding of the number sequence in this range. An illustration of this is that children can recognise a number of objects in some concrete scenario and can relate their knowledge of mathematical symbols and the number sequence as required.

Clearly a child's understanding deepens and expands throughout the foundation year, allowing the teacher to introduce addition at the start of the second term, thus building upon the skills developed in the first term. Due to their importance, number recognition, the number sequence and subsequently addition represent key foci for the teacher and thus a lot of teaching time is dedicated to them so that the children can practice applying their knowledge in different contexts and with different media to deepen their understanding. We can relate this to the tutoring system as it may be appropriate for the tutor to support one or more of these subjects.

The national curriculum specifies the order in which a teacher should introduce topics as well as specifying the amount of time a teacher should spend teaching them. It also recommends suitable activities that develop each topic in the form of games, practical activities, songs and so on, to encompass both mathematics and vocabulary usage. Therefore we can recognise that a computer-based mathematics tutor may have a "shelf-life" in terms of its usability throughout the foundation stage year. Clearly it should aim to support one or more topics well, rather than trying to support many reasonably well, and thus its applicability may only be for a particular period to support a given subject.

Further to the mathematics based lessons Louise Aylott also stated that an important lesson of the foundation stage is developing children's awareness of mathematics vocabulary so that they can effectively articulate the mathematics functions they are required to apply. Therefore we can infer that well articulated language and clear audible communication are of great importance in allowing children to practice:

- Listening to and interpreting instructions
- Listening to and subsequently using maths vocabulary in appropriate contexts
- Reasoning with a problem in some context and applying their knowledge appropriately

[10] Mapping a number to a set of objects for example

[11] Allowing a child to become proficient at counting forwards and backwards from any starting point in the sequence 0 to 10 for example

In relation to this both teachers emphasised the need for task instructions and feedback to be clear and concise so that the child can understand what they are required to do, and also what they have achieved. Speech and sound are important for this because they provide the means to mirror the mode of interaction that children are used to in the classroom. Moreover, HCI principles dictate that the human voice has the advantage of correct prosody and pronunciation and can therefore be easily interpreted. However, we can recognise that due to the non-permanent nature of sound output, the children should be able to repeat any audible communication for reasons of clarification.

As proposed by Louise Aylott, key words and pictures can be useful to identify the salient points of instructions or feedback, for example displaying the word yellow and colouring it yellow in an appropriate context. We can recognise that in such a case key words offer a reference to the user and can also aid their ability to accurately interpret information at the interface. However, we must consider the stroop effect. As described by Eysenck et al (1995) the stroop effect dictates that there is interference between perception and semantics in interface. Therefore if a user has to read the word "green" and it is written in red for example, there can be considerable user confusion and misinterpretation resulting in very long reaction times or user error. Thus we must consider the contexts in which key words are used in the interface so that they aid its interpretability and do not cause user confusion or detract from the interface's usability.

A notable need for the system that both teachers expressed was the need to inspire confidence in the child and their abilities. Clearly confidence is gained by getting things right and receiving feedback that justifies this, but it can be adversely affected by an overly complicated set of mathematical applications, or through an inaccessible interface that the child cannot understand or manipulate. As we have discussed in the literature review, confidence is key to motivation for users (Keller, 1983), but the teachers stated that confidence also aids children's development as it encourages uninhibited application of their knowledge and exploration of new concepts.

Both teachers recognised the need for a context or theme for the lessons provided in the tutoring system so that it emulates the teaching methods of the classroom. Much of the foundation stage involves initiating learning sessions where the child does not *know* that they are learning as such, hence the use of games, songs and practical activities that make abstract concepts accessible and fun for children. In a similar way to the "10 green bottles" song or the jigsaw game, the tutoring system should provide a focus for children in the form of visual prompts and audible sounds so that they can more easily conceptualise the task than if presented with abstract mathematical terms.

Computer-based education

All children are required by the national curriculum to undergo lessons in computer use when they join school. These lessons are delivered using a Whiteboard[12] initially, with subsequent sessions of child-computer interaction assisted and observed by the teacher. The aim of these lessons is to build a set of basic computer operation skills and also to raise the children's awareness and level of experience for interacting with ICT media. By discussing these lessons with the teachers it was possible to identify a generic skill set that all the children in the classroom have. Some of these skills were uncovered and recognised in the sessions of naturalistic observation (section 3.4.1), however through discussion we are able to clarify and reaffirm these skills. Thus the children have the ability to:

- Interpret sound and speech, and perceive information displayed on a standard-size computer screen, communicated in Standard English.

[12] A Whiteboard is a hybrid between a traditional whiteboard and a personal computer, and can essentially be described as a large touch-sensitive display on which the computer's image is shown, providing an ideal teaching tool for computer-based teaching sessions.

- Fully operate a mouse as an input device. This means that all children can comfortably manipulate on-screen objects via drag-and-drop, and also navigate interfaces through point and click operations.
- They do not have a well established skill base for using the keyboard as an input device. This means that we cannot use the keyboard as a method for interaction with the computer as we cannot assume that the children can use it independent of guidance.
- They do not have any experience of issues over multitasking and thus activities that require switching windows and simultaneous thought processes are not appropriate.
- They do not have any notable experience of using scroll bars to focus on key parts of the interface. All packages used in the classroom require the children to perform tasks in a fixed task domain.
- They do not have any experience of initiating a 'learning session' with the computer. For all computer packages used in the classroom the teacher must load the package prior to it being used by the children.
- The children are familiar with using software packages in the Windows environment. All of the packages used for tuition and recreation purposes in the classroom use a Windows operating system, and thus we can assume that the children are familiar with the look and feel of interface buttons, text and pictures.

This skill set provides a framework that the tutoring system must work within if it is to be suited to supporting mathematics education at the foundation stage.

3.4.3 Discussions with the Education Authority

All teaching media that enters the classroom is subject to evaluation by the education authority for early years teaching, who also recommend and buy certain products for classroom use. Discussions with Pat Lamb and Joyce Chester from Hampshire Education Authority for early years has enabled us to develop an understanding of the required role and salient features of tutorial software products.

Through discussion with Pat Lamb and Joyce Chester it was clear that the required role of a tutoring system is to provide opportunities for children to practice applying their skills in situations that extend those available in the classroom. They identified two main areas that children have to work through in early year's mathematics; visualisation and use of language, which usually require practical exercises or direct teaching if they are to be developed. In relation to this Pat Lamb stated that tutoring systems have been used in the classroom to offer the teacher another tool for developing these skills because they are highly visual, have strong animations and graphics, and use appropriate sound and speech. This form of tutoring is also beneficial to kinaesthetic[13] learners, which are common at the foundation stage, but it also forms an efficient method of conveying instructions, and developing children's understanding.

An important point made by Joyce Chester was that there is a need to ensure that children are not presented with ideas and concepts that are too abstract, or above their educational schema. This means that all classroom teaching aids, including ICT media must be simple with a clear relation to the objectives of the national curriculum. Through her experience of evaluating computer-based products, Joyce Chester emphasised the need for clear presentation of information and an uncluttered user interface that allows children to easily relate to the problems posed and manipulate objects themselves to arrive at an answer. From observation sessions in the classroom it was clear that direct manipulation of objects allows children to develop a better understanding of the "route to an answer", and this can clearly be directly applied to the computer-based tutor.

[13] Children whose preferred learning style is via the interpretation of important sounds and physical actions (i.e. they prefer a "hands-on" approach).

Pat Lamb stated that children learn primarily through their senses (sight, sound and feel mainly) by mapping their experiences to abstract concepts. As we have discussed in the literature review, it is the recalling of these experiences through sensorial stimuli that allows children to develop their understanding of mathematical concepts. Further to this Pat Lamb explained that ICT media provides the means for children with special needs to access these concepts. Highly visual displays are interesting and offer a different viewpoint on concepts with the advantage of holding the attention of children for longer than a book or worksheet.

Through Pat Lamb's experience as a teacher and through her observations of recent classroom lessons I was able to note the importance of teacher "assistants". Many teachers use a soft toy (for example a teddy bear) that works alongside them and essentially acts in a similar way to an interface persona. The toy helps the teacher to organise the children and provides a focus of interest and fun that the children can relate to. It can also be used to ask leading questions or provide feedback. Clearly we can map these benefits to the tutoring system through an interface persona to provide instructions and feedback in a similar fashion.

3.4.4 Discussions with Foundation Stage Software Designers

Discussions and interviews with 3 software designers, Jan Burd, Rod Boyes, Sue Rankin, from different software development companies has enabled us to gather more information on the problems faced when producing an educational software product. The following section details a summary of the more important points made.

All the software designers stressed the importance of making all classroom software as fun, engaging and interactive as possible. They also emphasised the importance of the tutorial content so that it is appropriately linked to classroom lessons. However, Jan Burd made the point that there is little value in producing a software product that the children don't find rewarding or interact well with, no matter how valuable its content is.
Furthermore the software designers agreed that it is advantageous to produce many tasks in a tutorial that cover the same educational strand in different ways and in different styles. Through discussion with the teachers and from the literature review it is clear that this is particularly salient as it enables children to link abstract concepts to many different situations. This enables a child to view concepts from different perspectives, which aids their understanding.

Clearly it must be considered that different children and indeed different genders respond to different types of stimuli and presentations and thus there is no universal solution to motivating or pleasing a child. However two important aspects that emerged from discussion were the need to use themed tasks and an interface persona because children can more easily interpret instructions, feedback and emotions through these media.
Moreover the need to allow customisability was noted as important by all the software designers. Sue Rankin in particular stated the need for the teacher to be able to access a set of controls over the difficulty of the tutorial, the colour of the interface and the sound for example.

We have already discussed the salient attributes of multimedia software for providing graphics and animation for use in a tutoring system. However, Jan Burd stated that graphics can be problematic as you need to ensure you correctly represent the scenario to the user. There is a need to be aware of the limits of a child's understanding, so not to overload them with information, or expect them to pick up on subtleties. This links with the view of the educational authority practitioners, who stated the need for simple graphics and uncluttered interfaces that are highly accessible.

Rod Boyes and Sue Rankin stated that the primary school market is the biggest for educationally supportive software, especially with the relatively new ICT requirements in the national curriculum. The need for children to use a variety of media for learning in the classroom and the potential for Information Technology (IT) to be integrated across the whole curriculum make tutoring systems popular with teachers and schools. Through 'content-rich' tutorials children can learn numerical skills, and at the same time fulfil requirements for developing their IT skills, regarding the use and control of computer hardware. Rod Boyes stated that previously Inclusive have provided so called 'content-free' software (for example painting packages) that solely caters for supporting IT as a separate subject to the rest of the curriculum. He stated that a much better approach is through computer-based tutoring software, for which good products are in great demand and are well received by schools.

A topic of particular interest when interviewing the software designers was the methods by which they produce their software products. Although all three designers used slightly different methods to produce marketable systems, there was a consistent theme of involving children and teachers throughout analysis and development phases. Iterative production methods were commonly used where a prototype is developed based on an initial specification. This is subsequently evaluated by child users and teachers from a number of representative schools, as well as other software designers to remark on faults, additions, short-comings and so on, before releasing a product. Sue Rankin stressed the importance of this process, stating that it provides the basis for fulfilling the true needs of the users, which relates to Carroll's task-artefact cycle (1991). Clearly we can map this mode of development to the tutoring system, such that user participation (teacher and children) can help to refine both the system's usability and the quality of the tasks that it presents.

The set of requirements gathered from these discussions, as well as all other stakeholder-centred requirements analysis activities are brought together below, with descriptions to justify their relation to the tutoring system.

3.4.5 Requirements gathered from Stakeholder-Centred Requirements Analysis

This section details the requirements gathered from all preceding stakeholder-centred requirements analysis activities.

Functional requirements

1. The system should present tasks using appropriate learning metaphors.
   <u>Description:</u> Learning metaphors should help to bridge the gap between the abstract mathematical concepts and concrete applications and thus encourage children to explore applying classroom taught concepts in interesting situations.
   Moreover, children should be able to indirectly develop their IT skills as well as their base of experience with ICT media by performing the tasks presented by the system.

2. The system should provide a set of tasks that 'change' to support the user's educational development.
   <u>Description:</u> The set of tasks should be *different* every time they are performed. The questions posed and data used to create mathematical scenarios should differ so that user is required to apply their knowledge and not just recall answers from memory.
   The set of tasks should incrementally increase in difficulty so to retain the child's concentration, provide a challenge and consolidate their learning.

3. The system should use graphics and animation to enable the presentation of highly visual tasks for the children to apply their knowledge within.
   <u>Description:</u> All tasks should be highly visual so that they are easy for children to interpret, provided they are accompanied by appropriate auditory stimuli.

4. The system should use speech and sound output as the primary mode of communication with the user.

Description: Foundation stage children have a limited ability to decode text and can more readily respond to speech and sound. From discussions with teachers we were able to recognise that children have the ability to interpret sound and speech, and perceive information displayed on a standard-size computer screen, communicated in Standard English. Moreover, audible communication mirrors the teacher-child interaction as seen in the classroom, and so should provide the user with some level of familiarity when using the system. It also provides the opportunity for children to practice listening to and interpreting instructions where mathematics vocabulary is used in appropriate contexts.
Coupled with the previous requirement for providing visually-rich tasks (requirement 3), we can also recognise that visual and audible communication media enables kinaesthetic learners to effectively use the tutoring system.

5. The user should be able to repeat all audible instructions and feedback.
   Description: Given that speech and sound are by their nature non-permanent, the user should be able to repeat the instruction set if they missed part of it or are confused by the message it conveys.

6. The interface should have an interface persona to direct and manage the task set.
   Description: A persona should be used as the source for communicating audible instructions and feedback. The persona should provide a focus of interest and fun for the children. The persona should pose questions and give feedback so to reflect the lessons that were observed in the classroom.

7. The system should have a reward scheme, providing a goal for the learning session.
   Description: A suitable reward scheme should be used to provide some motivation for the user to complete the task. It should also convey the user's achievements from completing the task.

8. The system should provide a printable summary of the questions answered.
   Description: Both the questions answered and the user's relative performance in the task should be summarised by the system, allowing a hardcopy to be printed for the purposes of the foundation stage profile.

9. The system should provide a set of teacher controls to allow appropriate customisation of the system interface and the tutorial content.
   Description: The teacher should be able to access a set of controls allowing them to customise the interface and the tutorial content.

Non-functional requirements

10. The system should be an effective teaching tool allowing children to practice applying mathematical concepts taught in the foundation stage classroom.
    Description: The mathematical content of the tasks presented by the system should be clearly mapped to the foundation stage national curriculum as well as classroom lessons so that concepts are not introduced above the child's schema.

11. The lesson length should be approximately 20 minutes.
    Description: The tutor should provide the teacher with an additional teaching tool that can be used as an activity from the 'job box'. In order to comply with other job box activities the lesson length should be approximately 20 minutes.

12. The system must be able to run on the on the classroom computers.
    Description: The system must comply with the following set of hardware requirements, sourced from the computing hardware available to the children at North Baddesley

Infant School. We have to assume that this is representative of the computing hardware that is available to infant schools in general.

- Microsoft Windows 1998
- Pentium 500Mhz processor
- 64Mb RAM
- An SVGA display capable of supporting 16-bit colour depth and a screen resolution of 800x600 pixels
- A graphics card capable of supporting 16-bit colour depth at 800x600 pixels
- A Sound card and a pair of headphones.

Usability and user interface requirements

13. The system should allow a child to work alone, without the need for teacher guidance.
    Description: Given the role as an activity from the 'job box', the tutor should allow a child to work alone, or in collaboration with other children, without the need for teacher guidance or support. The system should be suitably intuitive so that a child can interact with it and perform all the actions they desire without the need for third party interpretation or explanation.

14. The system should be able to be used in conjunction with the other computer-based tutors used in the classroom.
    Description: The system should complement the tasks and activities presented in other systems and thus not offer any conflicting messages that could potentially confuse child users.

User requirements

15. The mouse should be used as the key input device for interacting with the computer.
    Description: Child users can comfortably operate the mouse as an input device. Therefore appropriate point-and-click, drag-and-drop and selecting operations (direct manipulation) should allow users to perform all required actions and navigate the interface. This requirement relates to the system's usability as well as being a user requirement.

16. The interface should allow a large margin for error to compensate for the user's limited control over the mouse and thus the screen pointer.
    Description: The computing skills of the user should be considered throughout the design of the interface. There should be appropriately sized buttons that are well spaced to compensate for child user's limited control over the mouse as an input device. Buffer areas should be used to compensate for mistakes to minimise the chance of a user initiating an accidental action

17. The system should not require the user learn any new skills, and thus it should be tailored to their skill set and computing experience.
    Description: Through discussion and observation it emerged that the keyboard is not an appropriate input device due to the children's limited skills over its operation.
    Child users do not have any experience of issues over multitasking and thus activities that require switching windows and simultaneous thought processes are not appropriate.
    Child users do not have any notable experience of using scroll bars to focus on key parts of the interface as all packages used in the classroom require the children to perform tasks in a fixed task domain. Therefore the interface should not require the user to scroll to different parts of the interface.
    Child users do not have any experience of initiating a learning session with the computer. Consequently the teacher must load the software prior to it being used by the children.

3.5 Existing Tutoring Systems Evaluation

As mentioned in the introduction to this chapter, computer-based tutoring systems are not a new concept in the classroom. There are many systems on the market that aim to support foundation stage education in literacy, IT, art and numeracy providing children with a breadth of opportunities to practice classroom taught skills.
From analysing appropriate mathematics based systems we aim to uncover the advantages and well-designed features of the interface, interaction style and tutorial content that we can carry forward into the design stage. Any poor features that we identify will either be eliminated or if necessary their functionality will be redesigned.

A representative set of current tutoring systems have been selected, based on the tutors used frequently in the classroom of North Baddesley Infant School as well as an appropriate web-based tutor, recommended by teachers at the school. These are:

- A Teddy Bear's Picnic – Sherston Software
- Tizzy's Toybox – Sherston Software
- Numberacy lessons for 4-6 years – BBC website

A heuristic approach to evaluation will be taken with these systems to combine the usability and design principles suggested by Nielsen (2001) and Dix et al (2000) as detailed in the literature review. We will also include the issues that have arisen in previous discussions. However, as stated by Nielsen (2001) there are notable difficulties and disadvantages for an individual when performing usability evaluations alone, as one person will never be able to find all the usability problems in an interface. Thus to improve the effectiveness of the evaluation there was a distinct need to involve an HCI expert; Lucy Carruthers[14], thus reducing any possible bias or error.

Refer to Requirements Appendix section 1 for a detailed usability evaluation of the 3 tutoring systems, including relevant screen shots. The set of requirements gathered from this activity are detailed below.

3.5.1 Requirements gathered from Existing Tutoring Systems Evaluation

When evaluating the systems it was possible to note a number of well-designed features that we can consider further at the design stage. These design considerations are noted where appropriate with their relevant requirement.

Functional requirements

1. The system should display a complete set of the available task options and customisability options for the user.
   Description: The system should present a complete set of links for launching tutorial activities. Each link should have a picture that promotes user recognition of its function, as well as text to provide reinforcement so that the user does not have to recall the method necessary to navigate to the task they desire. In consideration of the user's potential lack of accuracy over the screen pointer (due to their control over the mouse), the link should also form a large clickable-area, instead of constraining the user to a small text hyperlink as demonstrated in the Tizzy's Toybox and BBCi tutors. Moreover these links should be well spread out across the interface to prevent the user from launching an activity accidentally.

[14] Lucy Carruthers is a lecturer for the faculty of Computer Science at Bath University. Lucy has built a wealth of experience in the Human Computer Interaction field of computing through work experience and a Masters degree studied at Bath.

Design Consideration: The system should have a main menu, with iconic buttons providing links to all tasks and all customisability options in one screen.

2. The system should use an interface persona to direct and manage the task set and provide an endearing figure for child users to relate to.
   Description: A persona should be used as the source for communicating audible instructions and feedback, as seen in all evaluated systems.

3. The user should be able to repeat all audible instructions and feedback.
   Description: Due to the non-permanent nature of speech output and its importance for communicating with child users for accessibility reasons, the user should be able to repeat any audible instructions or feedback on demand.
   Design Consideration: A button should be provided, that is clearly labelled allowing the user to repeat any audible instruction or feedback.

4. The system's mode of communication with the user should have a 'permanent' nature.
   Description: Further to the audible communication given by the interface persona (requirement 2), text should be used to provide some permanence to instructions or feedback at the interface.
   Design Consideration: The BBCi tutor also used text to reiterate all instructions and feedback, but the method used by Tizzy's toybox of printing only key words to screen, to reinforce the key point of the instruction is more appropriate given the literacy and decoding skills of foundation stage children.

5. The system should provide appropriate feedback to the user.
   Description: This relates to Dix et al's design principle of system responsiveness. The tutor should provide feedback to denote whether the user has answered a question rightly or wrongly, allowing them to gauge their level of their achievement. Quantitative feedback (for example "too few" or "too many"), as seen in the BBCi tutor was particularly advantageous as it provided the basis for the user to recognise and possibly reform their mistake. Quantitative feedback should be succeeding by the system allowing the user to have a second chance at attempting the question before revealing the correct answer if they answer wrongly.

6. The system should allow the user to control the task.
   Description: The system should comply with Nielsen's usability principle of user freedom and control so that the user is not 'locked-out' of the interface for prolonged periods whilst instructions are being given or demonstrations are being enacted (as seen in Teddy Bear's Picnic). Also the user should not be constrained to a strict procedure for completing the task without the option to quit a task at any time.
   Design Consideration: The user should be able to interrupt any instructions or feedback if they desire, and thus exert control over their progress through the task. This was an advantageous feature of the BBCi tutor.

7. The system should allow the user freedom to self-direct their learning session.
   Description: The user should be able to control the learning session by customising its level of difficulty to some degree, in pursuit of the development and extension of their knowledge base. Any incremental increase in difficulty should be carefully mapped to the child's mathematics ability and knowledge from classroom activities, as a concept like multiplication (as seen in the higher level of the BBCi tutor) is clearly above the educational schema of a foundation stage child.
   Design Consideration: When the user completes a task they should be able to choose whether they want to repeat it at the current level of difficulty, or move up to a higher level of difficulty.

8. The system should have a reward scheme, providing a goal for the learning session.
   Description: Further to system feedback, the reward scheme should give the child a local goal for the learning session that they strive to achieve. The reward scheme should also allow visibility of system status (a usability principle stated by Nielsen, 2002), such that the user can see 'where they are' in the task and thus assess their progress towards the reward scheme goal.
   Design Consideration: The BBCi tutor's gold coin reward scheme provided a local goal for completing the task, which is important for maintaining the user's motivation to complete the task. This reward scheme also provided visibility of system status as the user had to accumulate 10 gold coins to complete the task and thus could see how they were progressing.

9. The system should provide a printable summary of the questions answered.
   Description: The system should allow the user to access and print a summary of their performance in the task, once it has been completed. This was an advantageous feature of the Teddy Bear's Picnic and BBCi tutors.

10. The system should provide a set of teacher controls to allow appropriate customisation of the system interface and the tutorial content.
    Description: These controls should only be available to the teacher and not child users. The exit from the system should also be located in this area so that child users cannot accidentally exit the system during a learning session.

Non-functional requirements

11. The system should be capable of running as a standalone program.
    Description: A more general usability aspect gathered from evaluation was that the user should not have to download additional software to use or view any part of the system, as this requires an assumption to be made over the user's internet connection.

Usability and user interface requirements

12. The user interface should be easy to understand, operate and navigate.
    Description: The system should have a good navigational structure, so that it is predicable and so that navigational actions do not cause surprise to the user. The user should not be constrained from exploring the interface or the task and thus buttons should be provided to support their needs.
    In particular the BBCi tutor had a very poor navigational structure where the exit button from the task performed an inconsistent action in terms of where it took the user, reducing its propensity to be predictable.
    Design Consideration: The system should have clearly labelled iconic buttons that depict the actions they represent. Navigational buttons should enable the user to exit the task at any time and return to the main menu, and go forward to the next sum.

13. The user should not be able to click any buttons or hyperlinks outside the scope of the tutorial area
    Description: The user should not be able to navigate away from the system and thus become 'lost'. This was noted as a usability problem in the BBCi tutor as the BBC web-frame surrounded the task domain, with links to other domains. With the limited experience foundation stage children have of computer systems this is especially salient as they do not have the skills to navigate unfamiliar systems unaided, and thus undo accidental navigational actions.
    Design Consideration: The system should launch to fill the screen to prevent the user from clicking links outside the scope of the system.

14. The interface should be highly learnable, allowing users to easily pick up how to use the system.
    Description: The system should have a consistent colour scheme where colour pairings are carefully considered so that all text and pictures are easily legible. As seen in all evaluated systems, a consistent background colour should also be used throughout the interface to denote that the user is in the 'tutorial area'.
    The same font should be used for all text in the system, to avoid possible confusion.
    Further to this the interface should have consistently placed buttons for common actions (exit for example) and the persona should also operate in the same area of the interface, so to adhere to the predictability and familiarity design principles as stated by Dix et al (2000).

3.6 Requirements Validation

Through further discussion with Louise Aylott (a foundation stage teacher of North Baddesley Infant School) it was possible to validate the set of requirements to ensure that they are appropriate and suitable for supporting learning and teaching processes.
Louise stated that the requirements were generally satisfactory, although she did emphasise the need for her involvement throughout the design process so that she could adjudge the mathematical content and suitability of the tasks designed for the children, as well as the interface design. Louise also stated that she had concerns over three requirements:

- Functional requirement 3. Louise explained that she believed the task should not simply 'get more difficult' without the child being in control of this change. She stated that the child should be able to specify (using levels for example) if they want to attempt more difficult tasks. This enables them to have a greater level of control over their learning session, and also means that the system can more suitably support all children in the classroom, as some may not have the required depth of knowledge to attempt more difficult tasks.
- Functional requirement 14. Louise stated that she thought customisability options were not an important deliverable for the system, as she has never used these settings for any other educational tutoring system. From the evaluation of existing systems we saw that some of the systems allowed the teacher to control interface colours, the level of sound and the task difficulty. However, Louise explained that she did not require controls over colour and sound, provided that she was involved in the design process of the system. Also, in relation to functional requirement 3 (discussed above) Louise believes that the child user should be able to control the level of difficulty, making this redundant as a teacher control.
  Louise did however emphasise the need for the exit button (from the system) to be located away from the children's activities and tasks to prevent accidental error.
- Non-functional requirement 18. Louise stated that the use of headphones is essential when the system is used in the classroom especially if the main mode of communication with the user is via sound and speech, so not to distract other children.

A full specification of the requirements for the tutoring system is detailed in Requirements Appendix section 2. 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.

In the next chapter we will deal with designing the tutoring system in consideration of the requirements we have gathered. This chapter denotes the design solutions made to fulfil the requirements, as well as the decisions made that impact the implementation process.

4 Design

4.1 Introduction

We can recognise that from the point of view of child users that the user interface *is* the computer system, and thus the 'system image' provides the means for children to interpret the system's tasks, by means of buttons, animations, graphics and sound. Consequently the design stage involves making the system image explicit, intelligible and intuitive so that children are able to easily relate to it. Moreover, we must aim to make the functions of the system transparent so that users can understand what is being presented to them and what they must do in response.

In the pursuit of usability it is necessary that the users are involved throughout the design process. "It is not sufficient to interview users at the beginning for their requirements, and then to ask them to review the design once it is complete" (Moore et al, 1995), which is a point that was reinforced in discussions with the software designers. The reason for involving users and other stakeholders throughout the design stage is that the end system should be supportive of the tasks that children and teachers perform, rather than being convenient for the developer to implement. Moreover, when developing an educational tutoring system it is vital that the mathematical content is compliant with and complementary to classroom activities as well as the national curriculum, and thus involving teachers to provide guidance is of clear importance.

The following sections detail the design of the tutoring system where the aim is to produce appropriate solutions to the requirements that were specified in the previous chapter, and then iteratively redesign these with the input and participation of the users. Subsequently we can apply these solutions when developing a system that can be deployed and tested in the classroom.

4.2 High-level Design Decisions

At this stage it is necessary to make a number of high-level decisions to provide a scope for the system design and implementation.

Firstly, it is necessary to state that the implementation of the system will be a high-fidelity prototype, with the aim of embodying the knowledge gained from previous activities. Being high-fidelity by nature the prototype will provide complete functionality and be fully interactive with a clear definition of the navigational structure. Thus it will provide opportunities for users to explore and test as well as grasp an appreciation for its look and feel. Therefore the system only falls short of a full implementation because it will only offer the user one complete lesson; a so called 'vertical' prototype; "…an interactive, high fidelity prototype of only a subset of the product's available function" (Rudd et al, 1996).
When designing and implementing this prototype it is important that we do not simply reinvent what is already available, but also, given the previous explanation, it is clear we do not aim to create a system to seriously rival those seen in the evaluation of existing systems, due to its prototype nature. However, it is intended that the system improves on usability aspects of existing systems so that it is capable of effectively supporting the mathematics education of children at the foundation stage.

A second decision made at this stage details the mathematical content that the tutor will support. As the prototype solution will only provide one complete lesson in foundation stage mathematics, the mathematical content will be focused on just two of the main lessons from the foundation stage national curriculum. These are:

- Number recognition and the number sequence, and
- An introduction to addition

These subjects were stated as being the focus of mathematics teaching by the foundation teachers, as essentially they form the basis for all future mathematics education. This was reinforced through naturalistic observation of classroom teaching in the first school term, where it was noted that many learning metaphors and a variety of applications are developed to allow children to apply their knowledge of number and number sequence and thus become comfortable using them. Addition provides the first real application of these concepts, and is consequently an area for further focus in the second school term. Therefore combining the two disciplines in the lesson that the system provides has a natural mapping to the curriculum as well as classroom activities, provided that they are carefully explained and segregated into levels of difficulty.

A more informed design decision made at this stage is that the lesson provided by the system will be called Frog in the Pond, due to the learning metaphor that is used throughout the task. The rationale for using the 'frog in the pond' as a learning metaphor is explained and justified in section 4.6.1.

4.3 Prototyping

Prototyping is a very important stage in user-centred design as "it is often said that users can't tell you what they want, but when they see something and get to use it, they soon know what they don't want." (Preece et al, 2002). Accordingly we will use a low-fidelity prototype in the form of a paper-based representation of the system to give the user some visualisation of the planned interface.

Low-fidelity prototypes have limited functional and interaction qualities, but are particularly useful for depicting concepts, design alternatives and screen layouts (Rudd et al, 1996). Their purpose is concerned with aspects of design and not the mechanics of the tool and thus they can be used as the first step in proposing a fundamental design solution to the users. In this context we can develop an initial paper-based representation of the interface and its related tasks to depict the design solutions borne out of the requirements analysis process. This prototype can then be shown to and worked upon with foundation stage teachers and children to refine the interface structure, tasks, graphics and potential animations. From this we will arrive at a final design and specification that will be implemented in the form of a high-fidelity prototype, as previously stated.

The high-fidelity prototype will represent the core functionality of the final system such that we can deploy it in the classroom for testing and evaluation by children and teachers. Being high-fidelity we can draw realistic comparisons with existing systems, and also evaluate its success at complying with usability and design principles as well as fulfilling the requirements. It must be noted that high-fidelity prototypes require a significant development effort, similar to that of the implementation of a complete system, with only a lower cost in terms of time.

Having set out the terms of the design process, the following sections detail the development and refinement of ideas as directed by the process of iterative design.

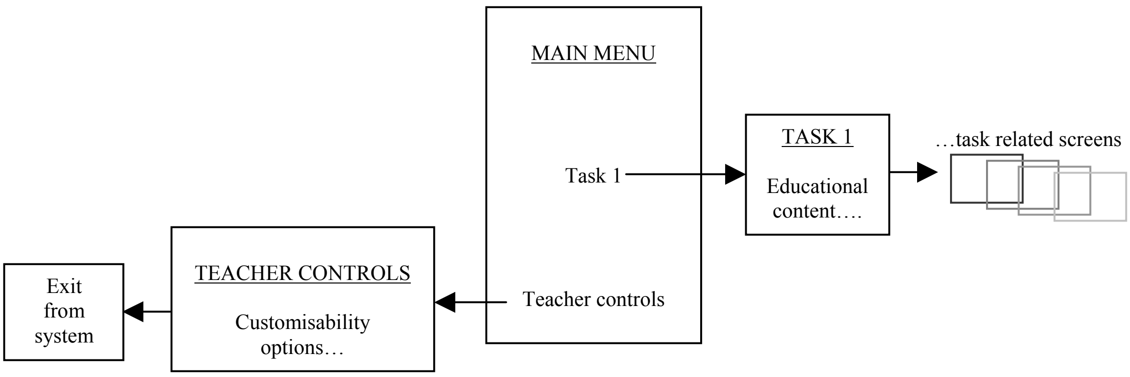
4.4 System Architecture

From the requirements analysis and specification chapter we were able to identify a number of functions that in theory could warrant having a screen of their own in the system's interface. However as Preece et al (2002) state, too many screens can be frustrating for the user and too few can cram-in too much information into too small a space. Thus a balance must be drawn.
From the evaluation of existing systems it was possible to recognise that Teddy Bear's Picnic and Tizzy's Toybox had a similar, well-designed system architecture that provided simplistic navigation for the user. From observing these systems being used in the classroom it was clear that their simplistic nature aided their learnability and usability, reinforced by the ease by which children were able to navigate their interfaces. By considering these designs in conjunction with the system requirements, we can identify the need to have 3 'zones' for user interaction:

- A main menu zone, providing a set of iconic buttons for all task options and all customisability options, in one screen. This relates to functional requirement 1.
- A teacher controlled customisation zone, providing a set of controls over aspects of the interface and the tutorial content. This relates to functional requirement 14. It has also been decided that the exit from the system should be located in this zone, so to prevent child users exiting the system by mistake. This was noted as a well designed feature of Tizzy's Toy box in the evaluation of current tutoring systems.
- A task zone, where each level of difficulty related to the task will have a set of related screens. This relates to functional requirements 2,3 and 4.

The 3 zones for user interaction form a fairly basic architecture for the system, although we can recognise that they provide the required functionality, reinforced by the relation of each zone to the set of functional requirements. Therefore we can define a high-level model of the tutoring system (refer to figure 4.1) to depict how the user navigates these zones, and how they interact.

Figure 4.1: High-level overview of the system architecture and interaction.



As is clear from this figure, it is necessary to define how the task related screens are structured which will be discussed in the following sub-section.

4.4.1 Task Structure

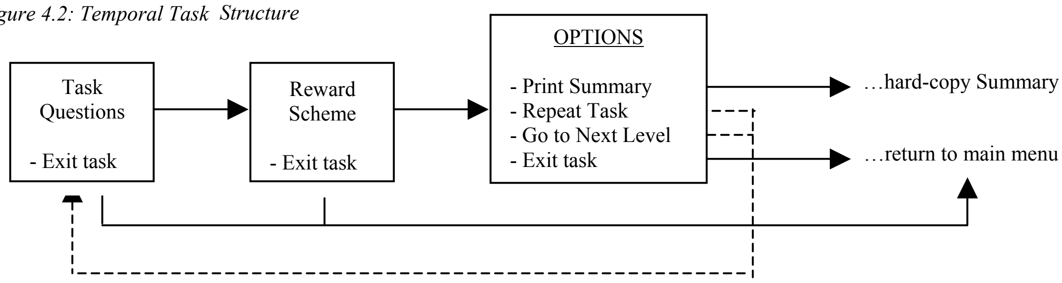
A logical format for structuring the task would be to allow the user to answer all the questions, receive feedback on how they have performed throughout the task, and then decide what they would like to do next (for example repeat the task, exit the task, and so on). Importantly, the user should also be able to exit the task at any time and return to the main menu, so to comply with Nielsen's (2001) usability guideline of providing an 'emergency exit'. This ensures that the user is not committed to the task or 'trapped', without having the opportunity to quit.

When evaluating Tizzy's Toybox and the BBCi tutor a very similar task structure was noted as a well-designed feature due to its local temporal format. Accordingly we have applied it as a design solution and developed it further so that a definite sequence of events can be defined:

1. The user launches the task using the button from the main menu, and is presented with a series of questions to undertake. A detailed explanation of the design of the task is in section 4.6.2.
2. After the last question of the task has been completed, the next screen details the task reward scheme, which provides feedback to the user on their overall level of achievement throughout the course of the task. Using a reward scheme to provide a goal for the learning session was cited as functional requirement 12.
3. The next screen presents a post-task set of options, which enables the user to choose what they would like to do next (relating functional requirement 11, which states that the user should self-directed their learning session). The set of options include:
   - Print a summary of the task for the purposes of the foundation stage profile (functional requirement 13).
   - Repeat the task at the current level of difficulty. In relation to functional requirement 3, the task should not reuse static data to pose questions and thus it should be different every time it is performed.
   - Go to the next level of difficulty. This also relates to functional requirement 3, such that the user should be able to choose whether they want to apply their knowledge in more difficult contexts and thus develop their understanding of the mathematic concept that the task supports.
   - Exit the task and return to the main menu.

This task structure will be called the 'temporal' structure, and is represented in the following diagram:

Figure 4.2: Temporal Task Structure



This structure dictates that the task is performed in sequence, and thus the user is able to consolidate and build upon the knowledge they apply in early levels when performing later levels[15]. However, we can also recognise that it may be restrictive for the user as it restricts them from attempting any level of the task at any time. That is, the user must perform level 1 before they attempt level 2, and they must complete level 2 before level 3 and therefore if they wanted to practice the harder concepts that are covered in level 3, or return to a previous learning session at a level other than 1, the system would not provide the flexibility to allow this. Therefore it has been decided that a second structure should be provided that the teacher can actively switch to so to provide this required flexibility.

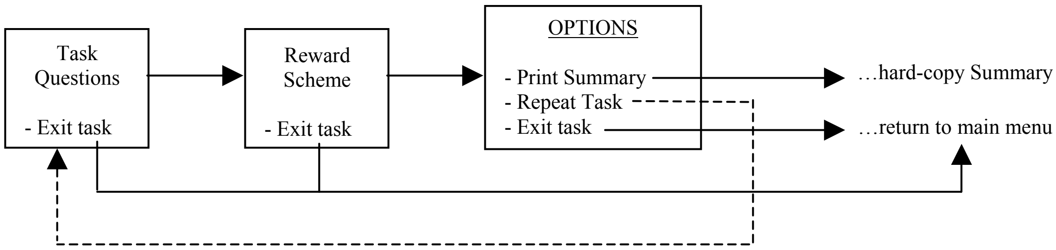
In relation to functional requirement 14 the teacher should have a customisability option for changing the appearance of the main menu so that buttons are provided to each level of the task; allowing users to perform any level at any time. So that the system is familiar for the user in both structures, the sequence of events for performing the task should not be *too different*:

[15] This relates to Hammond (2000) who stated that scaffolding is essential to learning, such that concepts should build in difficulty to consolidate previous learning and expand upon it.

1. The user launches one of the task levels using one of the buttons from the main menu, and is presented with a series of questions to undertake. A detailed explanation of the design of the task is in section 4.6.2.
2. After the last question of the task has been completed, the next screen details the task reward scheme, consistent with the temporal structure.
3. The next screen presents a post-task set of options, which enables the user to choose what they would like to do next. The set of options include:
   - Print a summary of the task for the purposes of the foundation stage profile.
   - Repeat the task at the current level of difficulty.
   - Exit the task and return to the main menu, where the user can click a button to initiate any of the 3 levels.

This task structure will be called the 'options' structure. Clearly, the key differences between this and the temporal structure is defined in the main menu (where there are buttons to each level) and at the end of level (where the user must return to the main menu to initiate the next level). This is represented in the following diagram:

Figure 4.3: Options Task Structure



Purposefully the questions of the task are performed in the same way as the temporal structure to promote the design principles of predictability and synthesisability (Dix et al, 2000). That is, changing the task structure using the teacher controls does not impact the questions of the task or their appearance in anyway, ensuring the user does not have to learn how to interact with the system under different circumstances.

Having defined two structures for the system we have clearly defined a framework for the system. However, there is an inherent need to consider the modes by which the user can input information to the system when acting within this structure, as well as the way the system will output information to the user.

4.5 Input and Output Considerations

The mode of user input and system output are very important aspects of the design as we must ensure that input is clear and easy to perform, and that output is easily interpretable.

To comply with usability requirement 20, it has been decided that the interface should use point and click operations to perform all navigational actions and perform all forms of user input. In relation to the method by which the user must enter their answer to a question, we can recognise that free input of answers would require a considerable task parsing system to evaluate the input and process it. Foundation stage children might also misinterpret the form of input required if a text box is presented within which they must enter their answer, and we can also recognise that this would be inappropriate given their computing skill base. Therefore limited or restricted answer input (multi-choice) is more appropriate and complies with the aforementioned need for using point-and-click input operations.

Newman et al (1995) recognise this style of interaction as graphical direct manipulation, which draws a clear relation to user requirement 29. This states that the mouse should be used as the key input device, and by nature all point-and-click operation must use the mouse. This style of interaction was apparent in all the systems that were evaluated in the requirements analysis chapter, although it must be noted that point and click operations should be used consistently throughout the interface. The Teddy Bear's Picnic tutor changed its style of interaction in relation to the level the user was performing, which was noted as being a major usability problem due to its implications of user confusion.

To comply with functional requirement 5, the mode of output should be audible in all zones for user interaction where the *child* must interpret information, give input or receive feedback. Clearly this must include the main menu, and all task related screens. The system should also present text keywords within these interface screens to provide a reference on the key points of the instruction or feedback, which was cited as functional requirement 8.
In relation to this design solution, it is possible to recognise that the use of language is very important for both audible and text output as we must ensure that it is easy to interpret and also that it is unambiguous. Language must also be very closely related to the classroom so that children are not confused by the messages output through the interface. Accordingly the design of all system output must involve the foundation stage teachers and also encompass the knowledge gained from observational sessions performed during the requirements analysis stage. However, given that the output will vary according to the context in which it is used (for example instructions given at the main menu will differ from those given in the task) we will deal with the specifics of the use of language in the detailed design of the system, in section 4.6.
In the teacher controls area of the interface, audible output is inappropriate as this section is intended only for teachers. Therefore simple text instructions should be used to denote the tasks that the teachers can perform.

4.6 Detailed Design

This section provides the essential details of the system design to produce design solutions to the requirements and also to produce the low-fidelity prototype that will provide the basis for user-centred discussion and redesign.

4.6.1 Specific Features of the Design

In the requirements document we specified three specific aspects of the system that were of particular importance if the system is to be usable for foundation stage children and truly suited to supporting their education:

- Requirement 2: the need to use an appropriate learning metaphor.
- Requirement 7: the need to use an interface persona.
- Requirement 12: the need to have a reward scheme to provide a local goal for completing the task, as well as provide some visibility of system status.

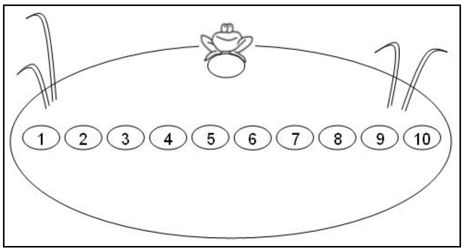
The rationale for stating that these designs features are important is that they have a significant effect on the way information is presented and conveyed through the interface as well as an impact on the design of the system as a whole.

Learning Metaphor Design

As stated in section 4.2 it has been decided that the 'frog in the pond' will be used as the task learning metaphor due to the frequent use of animals in a variety of classroom activities as props for learning. Accordingly we can infer that the children are familiar with them in a learning context. Moreover the children at North Baddesley infant school have access to a pond in the school grounds and thus have experience of seeing frogs in their real world habitat[16]. This relates to Richmond (1970), who stated that "all new experiences must be related to experiences which children already understand." Therefore we can recognise that the frog in the pond metaphor provides a concrete scenario for children to apply classroom taught abstract concepts, with a clear relation to their interests and experiences.

The frog will inhabit a pond with 10 lily pads, each of which will be button that is clearly labelled with a number from the number sequence in the range 1 to 10 (refer to figure 4.4). These buttons will form the mode of user input to answer the questions.

Figure 4.4: Frog in the Pond learning metaphor graphic



All parts of the task will revolve around the sequence of lily pads, providing an application for children to calculate their answers. The frog will jump between the lily pads to reinforce aspects of the task instruction, and also to demonstrate the correct answer as feedback. As observed in the classroom, perceptual prompts were used successfully in songs and games[17] and it was noted that this aided children's understanding because they could relate to the changes in the environment more easily than having to think in abstract terms. This can be reflected by the frog moving between the lily pads as part of the task's learning metaphor.

Visualisation is a key feature of this metaphor, such that the children should be able to perceive a pond, a set of lily pads and a frog, and when used in the context of the task they should be able to relate these objects to the mathematical concepts of number, the number sequence and addition as required. This relates to Piaget's summation that children in the "intuitive phase" of cognitive development are dominated by sense of visual perception (Copeland, 1979), and thus it is possible to recognise that this learning metaphor is relevant for them. It must also be noted that due to its relatively simplistic nature, the 'frog in the pond' can be accurately represented using graphics and animation in the tutoring system so that it is readily interpretable. This is important if the previous discussions regarding the benefits of the learning metaphor are to be supported.

Interface Persona Design

The need for using an interface persona is to direct and manage the task and thus provide all audible communication to the user. This was cited as functional requirement 7. It was decided that a 'teacher-style' figure should be used for the persona, to convey the importance of the information that it provides and also to reflect the adult-child mode of interaction enacted in the classroom. It was also important to use a personable character for the children to relate to, as they must work *with* the persona throughout the task.
Given these considerations, a male teacher graphic was developed as shown in figure 4.5.

[16] Having access to a pond in the school grounds is a common feature of many infant and primary schools and thus we can recognise that the premise of children having experience of a frog's habitat is upheld in other schools.
[17] The 10 green bottles song for example where children sat down from a standing position to depict 'one less'.

Figure 4.5: The interface persona graphic

In the high-fidelity prototype this graphic will have an animation to show that its mouth is articulating the words that are output as part of the system's audible communication. This is an important design feature as users must be able to identify the source of all instructions and feedback so to uphold functional requirement 7, which states that the persona should direct and manage the task set.

Reward Scheme Design

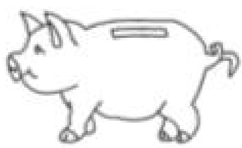
In the evaluation of the BBCi tutoring system we were able to recognise that awarding children a gold coin for each correct answer they gave to a question was a particularly well designed feature because it gave users a local goal. Accordingly we have applied this design solution and developed it further.

The system will award the user with a gold coin for giving a correct answer to a question; with the aim of accumulating as many gold coins as possible throughout the course of the task. In relation to functional requirement 12, these coins will be accumulated in a 'progress indicator' at the top of the screen, where there will be a 'box' for each question. When the user gets a question right they are awarded a coin and this is placed in the relevant question-box. Therefore the progress indicator provides some visibility of system status (usability guideline, Nielsen 2001), as the user can see how many questions they have attempted, how many they answered correctly and how many they have left.

At the end of the task the total number of coins will relate to an animation, where a higher total is rewarded with a more elaborate animation, thus providing a sense of fun and also denoting the user's level of achievement. Moreover it provides a sense of closure, which Dix et al (2000) states as being important so that the user can appreciate that the task is finished, and thus prepare for the next task.

In consideration of the use of gold coins as a reward for correctly answering a question, it was decided that the metaphor of using a piggy bank as the figure to perform the final animation would be appropriate and would provide the required sense of fun (refer to figure 4.6).

Figure 4.6: Piggy bank graphic: the reward scheme final animation



We can recognise that collecting coins in a piggy bank is familiar for the children due to its relation to the real world experience of saving money; therefore it should be readily interpretable. This metaphor also continues the theme of using animals as predominant figures in the tutoring system, which draws a further parallel with the activities of the classroom (where animals are used extensively as learning props).

The application of these three specific design features to the overall system is detailed in the following sections.

4.6.2 Task Design

A key part of the system design is the task that it provides, as clearly we must carefully consider the methods by which mathematical concepts are embedded so that they can be easily accessed for practice. As already stated, we have decided to support the concepts of number, the number sequence and addition. It has also been decided that these concepts should be divided by levels of difficulty to reflect the fact that addition is a harder concept than the number sequence[18] and also to segregate these concepts. Moreover, this reflects the order by which these concepts are taught in the classroom, as well as the order in which they appear in the national curriculum.

Levels 1 and 2

Levels 1 and 2 of the task provide the opportunity for children to apply their knowledge of the number sequence as well as basic addition. These levels are meant to directly reflect the 'number in the bag' lesson that was observed in the classroom. This lesson had an appropriate mathematical content that required the children to apply their knowledge of the number sequence in a fundamentally different way from simply counting from 1 to 10. It was noted in the session of observation that this challenged the children as they had to apply their knowledge from unfamiliar starting points in the number sequence and subsequently perform simple calculations over addition (for example adding 1) to further test their understanding. Therefore we can recognise that it is particularly suitable to reflect the key points of this lesson for the purposes of levels 1 and 2 of the task, where level 2 will simply be *more difficult* than level 1. In-keeping with the format of the classroom lesson the task will have the following format:

- A starting number will be presented (for example 3) and subsequently the interface persona will give the instruction that the child must find the number in the number sequence that is 1 more than it (level 1) or two more than it (level 2). The use of language for this instruction will be very closely related to the classroom lesson to ensure that it is both appropriate mathematically, and to ensure that it is familiar for the children.
- The child user must then calculate their answer (using the lily pads as the application for arriving at their answer for example). Each lily pad acts as a button for user input, where the user has to click on the one that they think relates to the correct answer (for example if the answer is 4, the user would click on the lily pad labelled 4).
- The interface persona can then use appropriate audible feedback to direct the user on whether they have given the correct answer or the incorrect answer (relating to functional requirement 9).

Refer to Design Appendix section 1 for a more detailed explanation of level 1 and 2 of the task (including screen mock-ups).

Level 3

This level of the task provides the opportunity for children to perform basic sums of addition. This is a suitable follow-on to the previous two levels as they provide a grounding in the concepts of number and the number sequence, thus priming the children for a more difficult application of their mathematical knowledge in this level. Level 3 has been designed to mirror the lesson in addition that was observed in the classroom such that children are presented with a sum (for example 3 + 3) and are required to use physical props to calculate their answer (for example they count 3 objects, and then count another 3 to form a total). Clearly we can replicate the use of physical props in this context by using the frog in the pond learning metaphor, and thus allowing children to count lily pads in a similar fashion.
In-keeping with the format of the classroom lesson the task will have the following format:

[18] Hence addition is a lesson of the second term in the foundation year, while the number sequence is a lesson of the first term.

- A sum over addition will be presented (for example 3 + 3) and subsequently the interface persona will give the instruction that the child must find the total. The use of language for this instruction will again be very closely related to the classroom lesson to ensure that it is both appropriate mathematically, and to ensure that it is familiar for the children.
- The child user must then enter their answer by clicking on the relevant lily pad in the sequence (for example lily pad 6 in the example 3 + 3). This mode of input is identical to the previous two levels, thus promoting the design heuristics of predictability and synthesisability (Dix et al, 2000), such that the user can apply their experience of interaction from the previous levels.
- The interface persona can then use appropriate audible feedback to direct the user on whether they have given the correct answer or the incorrect answer (relating to functional requirement 9).

Refer to Design Appendix section 1 for a more detailed explanation of level 3 of the task (including screen mock-ups).

The 3 levels present suitable scenarios for children to practice applying mathematical concepts because they are very closely mapped to lessons of the foundation stage classroom and thus the national curriculum (clearly relating to non-functional requirement 15).
We can also recognise that by reflecting the salient aspects of the classroom lessons the task should be familiar for the children thus helping to reduce learning time for the user when using the system to apply their knowledge.

General Aspects of the Task Design

A more general decision made at this stage is that each level will consist of 8 questions. This design decision was dictated by the fact that level 2 of the task by its nature can only pose 8 different questions[19]. However, 8 questions per task is the same length as the majority of the tasks evaluated in the evaluation of existing systems and it was also approximately the same number of questions asked by the teacher before she either incremented the level of difficulty or ended the lesson (i.e. after 8 questions the children will be comfortable at performing the task and could be tested in more difficult contexts). Therefore we can recognise that it is an appropriate number of questions. Moreover, setting 8 questions per level provides consistency and predictability over the format of the task.
In relation to this, the questions for each level should also follow a temporal structure, such that they are presented one at a time, and each one should be completed before the next can be undertaken. This prevents user confusion and complies with user requirement 31 as it means that users do not have to consider multitasking (of which foundation stage children do not have any experience). This design solution also relates to the reward scheme's progress indicator, as clearly where the child attempts a question, gives an answer and gets feedback on their answer they can subsequently prepare for the next question where the procedure will be repeated.

4.6.3 Teacher Controls

As already stated, it was decided that the teacher should be able to switch between the 'temporal' or 'options' structure in order to change the display of the main menu and thus support the variety of needs that are apparent in the classroom. Other customisability controls that were apparent in the evaluated systems allowed the teacher to change the volume of audible output and the colour of the text displayed through the interface. However, volume control is redundant due to the control on the computers speakers, and the colour of the text should not

[19] Level 2 requires the user to find a number that is 'two more' than a starting number, and given that the lily pads support the numbers 1-10, only 8 different questions can be posed.

need to be changed if it is designed appropriately in the first instance, and thus these can be discounted as being relevant to the system. Therefore the only customisability option available to the teacher will be over the 'task structure'.

As already stated, it was decided that the teacher should be able to exit the system from the teacher controls screen to prevent child users from accidentally closing the system during their learning session. There should also be a 'yes/no' confirmation when exiting the system, so to provide the opportunity for the teacher to undo this action if necessary.

4.6.4 Low-fidelity Prototype

The initial design of each screen of the interface and its navigational structure is detailed in Design Appendix section 1. This design forms the low-fidelity, paper-based representation of the system that has been developed in consideration of the framework dictated by the system architecture, the detailed design, the task design and the requirements.

4.7 User-centred Discussion and Redesign

After producing the initial low-fidelity prototype design of the system, it was necessary to hold discussions and sessions of feedback with the users in order to refine aspects of the design and address any issues they felt were important. Only the pictorial representation of each screen of the interface was presented to user and explanation and questioning were used to probe salient aspects of the design to gauge the user's opinion.

The discussions were performed with Louise Aylott and Janis Reeves, both foundation stage teachers. These teachers both partook in the requirements analysis and specification, and thus have a detailed knowledge of the project and its aims, hence their suitability for this activity. Moreover, involving two teachers provided a dual-aspect on assessing the appropriateness and success of the initial design.
It was decided that testing the low-fidelity prototype with child users was inappropriate after attempting to discuss the design and perform interviews with a number of children in the classroom. It was possible to recognise that the children could not easily interpret specific aspects of the design from the paper-based representations, especially when told that they were directly mapped to a computer interface. Moreover, it was noticeable that the children could not form a detailed critique of the prototype reinforced with any justification. Thus the foundation stage teachers were the main source for the user-centred discussion, and where appropriate they gave feedback on whether they believed children would appreciate the design, given their teaching experience. The key points of the discussion are detailed below, with the related design solution, where appropriate:

Louise Aylott thought that the 'fake' buttons in the main menu, which are used to denote areas where other lessons can be added in future system extensions, should be removed for the purposes of the prototype. She believed that they could cause confusion for the children, who may think that these are actual buttons with related actions.
<u>Design Solution:</u> The 'fake' buttons were removed from the main menu.

Both teachers reinforced the need to use roll-over button effects to give the user some feedback to denote that they are 'on' them (i.e. a roll-over effect). Louise signified the importance of this with children as children could interpret iconic buttons as just pictures embedded in the interface, and thus not realise that they represent navigational actions for example.

Both teachers liked the 'frog in the pond' learning metaphor and believed that it was appropriate for the children. Janis Reeves stated that the natural structure the lily pads provide is well suited to the mathematical concepts of the number sequence and addition because they reflect the "hop-by-hop" nature of counting and also provide individual objects that the children can count when performing calculations over addition. Moreover Janis said that it was important for the interface to have some fun appeal directed through a character such as the frog, so that the children have some reason and motivation to follow the task and complete it.

Louise stated a concern over the design of levels 1 and 2 of the task (Design Appendix section 1, figure 3 and 4). Louise thought it was appropriate that the activity was closely related to the 'number in the bag' lesson that is used in the classroom, however she thought that it should be redesigned to relate to this lesson with even greater accuracy, so to further promote the children's recognition of the classroom based lesson.
Design Solution: Louise suggested that the focus of the task should be redesigned to be the number, instead of the frog. For example where the question is "what is 1 more than 3", Louise thought that 3 should be the focus of the task. She stated that the lily pads should then form the application for calculating the answer, so that the child can count them as individual objects as they do in the classroom.
The previous design of levels 1 and 2 involved the frog being positioned on the relevant lily pad (3 for this example) and thus the user making a trivial calculation on the required answer (for example 4; the lily pad next to the frog). This clearly reduced the need for children to really apply their knowledge. The redesign addresses this issue, as directed by Louise.
Refer to Design Appendix section 2, figure 1 for the screen design and further explanation.

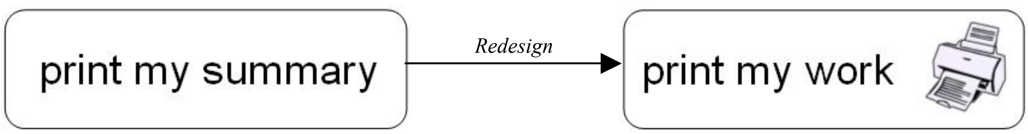
Both Louise and Janis stated the importance of using appropriate mathematical language in all levels of the task, so to mirror the language used in the classroom. Louise said that she was happy with the language used in level 1 and 2 as it directly reflected the 'number in the bag' lesson used in the classroom. However Louise and Janis suggested a slight change to the language used for level 3, where the child must perform calculations over addition.
Design Solution: Louise and Janis said that children are taught to articulate an equation such as "3 + 3 = 6" as "3 add 3 *is* 6" instead of the anticipated "3 add 3 *equals* 6". Therefore we can redesign the audible output for level 3 to comply with this:

- Feedback (if correct): "Yes that's right! 3 add 3 *is* 6."
- Feedback (if wrong for second time): "No, actually 3 add 3 *is* 6."

Louise and Janis both thought that the piggy bank animations that form the final animation for the reward scheme would be well received by the children. Janis stated that the animation would provide some interest and would be enjoyed.

Janis agreed with the simplistic nature of the process for printing a summary of the task from the 'end of game' screen (refer to Design Appendix section 1, figure 7). However she explained her concern over the label of this button as she thought that children would not understand the word summary in this context. Therefore she suggested it was renamed to be more intuitive.
Design Solution: The 'print my summary' button has been renamed to 'print my work' and also carries an icon to denote that their work is printed as a result of clicking it (refer to figure 4.6)

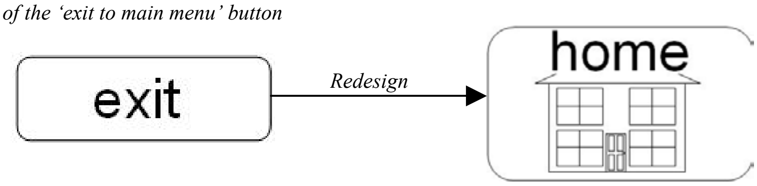
Figure 4.6: Redesign of the 'print summary of task' button



Janis also thought that 'exit' button from the task was a potential point of misunderstanding for the children, as it does not represent *where* they are taken by clicking on it. She thought that many children would not understand the word exit in this context.

Design Solution: Janis suggested that the button was renamed 'home' (in reference to the main menu) and she thought that it could have an icon with a picture of a house to reinforce its navigational function, see figure 4.7. Clearly this redesign is particularly suitable for the system because referring to the main menu as 'home' would suggest that it is the centre-point of interaction, which of course it is.

Figure 4.7: Redesign of the 'exit to main menu' button



The redesigned 'return to main menu' button can be applied to all areas of the interface that are appropriate, including all task related screens, the reward scheme screen, the end of game screen and the teacher controls screen.

By discussing the button redesign it emerged that there was a need to redesign the main menu in accordance with it. Therefore the main menu's title 'games' was changed to 'house of games', and a 'house' metaphor was applied to define a new structure to the main menu (refer to Design Appendix section 2, figure 1 for the screen design and further explanation). This redesign significantly improves upon the previous design as it defines a 'location' from which the child can initiate their learning session, as well as a location to return to when they finish a task. Moreover it may help to promote the system's learnability, as both the house metaphor for the main menu and the home metaphor for the navigational button promote recognition of their functionality.

4.8 Conclusions

Having produced an initial design and refined this in consideration of the user's needs and desires we can formulate a final specification of the design solutions to the system requirements (refer to Design Appendix section 3).

With reference to the design solutions it is clear that usability requirement 21 has not been met. This states that the interface should allow the user to directly manipulate objects for relevant tasks, so to provide a different perspective on how to arrive at an answer in a mathematics context. This requirement has not been upheld as it was decided that point and click input operations should be used to allow the user to select their answer to all questions, thus supporting the multi-choice format. Moreover the lily pad objects that form the learning metaphor are meant to be used as an application for calculating the answer. This was designed to reflect the lessons that were observed in the classroom where physical props are used to assist the children when calculating their answers. Accordingly it was decided that these objects should not be able to be manipulated by the user.

It is also necessary to state that non-functional requirement 17, which states that the system should not produce any unexpected behaviour, will be enacted through system testing so that any errors are eradicated. Also non-functional requirement 19, which states that the system should be a standalone program will be fulfilled in the implementation chapter, where an executable file should be delivered for launching the system.

The design solutions specified to meet the requirements can clearly be directly applied when implementing the high-fidelity prototype in the next chapter. This chapter denotes the essential details of the implementation of the system, which should subsequently be deployed and tested in the foundation stage classroom.

5. Implementation

This chapter details the implementation of the high-fidelity prototype that will provide the basis for evaluation in the foundation stage classroom. Throughout this chapter we will apply the design solutions that have been specified in the previous chapter.

5.1 Implementation Language

In both the requirements document and the design chapter we discussed and specified the need to provide a graphics intensive high-fidelity prototype with audible output and direct manipulation interactivity. Given this specification there are only a limited number of software packages and programming languages that can be used for developing the solution. These include:

- Macromedia Flash with ActionScript programming language
- Java and Java Applets
- C++
- A web-based program developed in HTML with a database back-end

After consideration of these implementation methods, it has been decided that Macromedia Flash MX 2004 with ActionScript will be used because it is ideally suited to supporting the production of highly interactive computer programs (refer to Implementation Appendix, section 1).
Flash was preferred to Java and C++ as these programming languages have a very large programming overhead and thus development time, and although they support the production of graphics intensive applications they are not purpose-built for this function. Moreover both these programming languages require an appropriate compiler and virtual machine to be installed before they can be run from the classroom computers which would be inappropriate given the specification of non-functional requirement 19 (the prototype should be a stand-alone program and not require any addition software to be downloaded for it to run).

5.2 System architecture

As the designed system architecture dictates, there are a number of 'zones for interaction' in the system. In accordance with this and to comply with 'good flash programming practice' a separate flash file for each zone will be created, meaning that when the final system is published it will be possible to run just 1 file and then 'load' and 'unload' files as required (in response to user navigation). This means that the cost to the system memory and processing time is minimised, given that each file by nature will contain detailed graphics, animation and sound.

Therefore we can define 10 flash files that need to be created:

Figure 5.1: Table to illustrate the flash files that form the system architecture:

	Temporal Structure	**Options Structure**
Main Menu	tutor_MAIN.fla	tutor_MAIN_options. fla
Teacher Controls	tutor_teacher_controls. fla	tutor_teacher_controls_options. fla
Level of task:	tutor_level_1. fla	tutor_level_1_options. fla
	tutor_level_2. fla	tutor_level_2_options. fla
	tutor_level_3. fla	tutor_level_3_options. fla

By specifying each of these files as separate entities we also gain the advantage of being able to copy and reuse salient aspects of them. Library components, ActionScript coding and file properties are all inherited from the master flash file when copied to produce a new file. Aspects of this new file can then be modified as required. Clearly this defines a more efficient method for implementation than if these files were to be created individually. Accordingly we need actually only create 4 flash files and then reuse them to create the other six:

- `tutor_MAIN`: This can then be copied and modified to produce `tutor_MAIN_options` where we need only change the buttons for accessing the levels of the task.
- `tutor_teacher_controls`: This can then be copied and modified to produce tutor_ `teacher_controls_options` where we need only remove the button for changing to the options structure, and replace it with a button for changing to the temporal structure.
- `tutor_level_1`: This can then be copied and modified to produce `tutor_level_2` and `tutor_level_3`. The procedure for modifying these files after copying is more complex than with the main menu and teacher controls due to the need for changing the underlying ActionScript code, although all library components and frame settings are inherited.
- `tutor_level_1_options`[20]: This can then be copied and modified to produce `tutor_level_2_options` and `tutor_level_3_options`. Again, the procedure for modifying these files after copying is more complex than with the main menu and teacher controls, as was detailed with `tutor_level_1`.

As already stated, each of these files will be only be loaded in response to a user navigating to a different zone of interaction (for example from the main menu to the teacher controls). When loading and unloading files in flash the `loadMovie()` function can be used. This allows a movie file to be loaded into a 'layer' of the flash player, allowing a scene to be constructed with many movies running concurrently. However, to ensure that memory nor processing time is wasted, we can also utilise the `loadMovie()` function to load movies into the same layer, with the effect of replacing the movie that is currently running. This is ideal for the purposes of the tutoring system, as user requirement 30 states that the user should not have to deal with multitasking, and thus running a movie on a layer underneath another would be redundant anyway. Moreover by closing a movie and loading another, the user can interrupt the interface persona or quit a task without the play-head affecting the loaded movie or confusing the output.

5.3 Setting the Properties for Flash Files

Flash files are edited as '.fla' files, within the Flash development environment (refer to Implementation Appendix, section 2). Given the system architecture and the number of flash files that need to be created it is necessary to set a number of properties on all the flash files so that they are consistent in appearance. By carefully implementing consistent features of the movie stage[21] and the interface scene[22] the quality of movie playback is also improved, as the movie playbacks runs more smoothly.

5.3.1 Interface setup

- The size of each screen of the interface must comply with that of the classroom (non-functional requirement 18) and thus the stage for each screen of the interface is set to 800x600 pixels.

---

[20] This flash file can actually be copied from `tutor_level_1` initially, and then modified before it is reused to produce levels 2 and 3.

[21] The stage if the area of the interface that is visible to the user. Essentially if the movie is projected to full-screen resolution, the stage is the area that falls within the bounds of the screen dimensions.

[22] A scene describes all the components and code relevant to a particular screen in the interface.

- The frame-rate is set at 12 frames per second. This is stated as being "an optimal frame rate" for playing flash animations where a trade-off has been made between quality of playback and the size of the flash file (Macromedia, 2004).

5.3.2 Font

The `_sans` font has been used for all text in the interface as this is platform independent and thus will be rendered consistently and predictably when performing all tests. Moreover, `_sans` is a very precise and unambiguous font and thus should be easy to read and interpret for the children.

5.3.3 Component Library

The component library can be used to store the 'master copies' of graphics, animations, buttons and sounds, where any number of instances can be used in the flash file, or imported to other flash files. It is good flash programming practice to save all interface components into the library and also to organise it into folders that group these components according to the particular frame or interface scene that they belong to. This was complied with throughout the implementation process.

Similarly to the development procedure specified for the flash files that constitute the system architecture, we can copy, reuse and modify components to produce similar components. This is an efficient method for implementation as attributes can be inherited from the 'master copy'. This was particularly important when implementing the animations (see section 5.4), as it was possible to copy a library component and then modify individual frames or the attached sound (for example) in order to create a new component.

5.4 Creating Components

Before creating each of the movies specified in section 5.2 it was necessary to create all the components of the component library. By doing this each movie can inherit the same interface components, thus promoting their consistency and familiarity (Dix et al, 2000).

5.4.1 Sound

All audible sounds were recorded using Microsoft Sound Recorder v5.1. This is an appropriate software application for creating short '.wav' files sound clips, which can subsequently be attached to an animation using an 'invisible sound' layer.
In order to record the required audible sounds in the most efficient manner and thus reduce a) the size of flash files they are embedding within and b) the number of recordings to perform, individual words or phrases were recorded and concatenated to support reuse. For example:

- "what is" + "one more than" + "one"
- "what is" + "one more than" + "two"
- "one more than" + "one" + "is" + "two"
- "one more than" + "two" + "is" + "three"

In the example given we have 4 sentences and only 6 different '.wav' sound clips.

5.4.2 Buttons

All buttons in the interface have a 3-dimensional appearance to give the impression that they are manipulatable and are not just embedded pictures in the interface. Flash provides excellent support for creating button effects, such that it is possible to draw the button graphic and then specify 4 separate frames to support each of the following button effects:

- Up: All buttons have clear text labels and icons (if appropriate) to promote recognition of their function.

- Roll-over: The text label for each button is highlighted[23], to make it obvious that the screen pointer is over the button hotspot.
- Down: All buttons invert their colour when down, to feedback that they have been acted upon by the user. For example a dark blue button with a light blue border is inverted to light blue with a dark blue border.
- Hit: After being hit all buttons return to their 'up' status.

When embedding in a scene, a button can also have ActionScript attached to it in order to specify conditions for the actions launched on its release:

Figure 5.2: ActionScript for specifying button conditions

Using this function it is easily possible to attach navigational actions to buttons (for example `loadMovie()` to open a different movie file) as well as other user defined actions.

5.4.3 Graphics

Graphics are created in flash using the graphics development tools that support freehand drawing, the construction of shapes and lines and also the creation of a variety of fill colour effects. It is possible to create graphics using layers so that salient aspects can be separated and also laid on-top of each other in order to build more complex images. Examples of graphics development during the implementation phase are the interface frames that are used throughout the system. There are three types of interface frame to support the display of information relevant to the particular zone of interaction.

- `Interface_frame`: This is the general use frame for all areas of the interface that the children interact within and thus includes a frame for the persona.
- `Interface_frame_task`: This is the frame for all questions of the task and the reward scheme. This frame is the same as Interface_frame, but also includes the progress indicator.
- `Interface_frame_teacher_controls`: This is the frame for the teacher controls area, and does not include a frame for the interface persona or the progress indicator.

Clearly only 3 graphics needed to be created here and then saved to the component library. Subsequently instances of the appropriate frames were used in relevant movies to create a consistent appearance.

5.4.4 Animations

Animations are instances of graphics that are converted to movie clips. All layers associated with a graphic are maintained, but the component is given its own timeline for the movie play-head to play through, within which the animation must be specified. Therefore unless the clip is designed to repeat itself, it must have a frame at the end of the timeline with `stop()` attached to it.

'Self-authored' animations can be created by specifying all frames between a starting point and an end point on the movie timeline such that all parts of the animation are defined by the author. However, flash also has two animation functions that support the animation of a graphics and the distortion of shapes. These functions (shown below) can be combined and manipulated as required to produce a range of animations:

[23] An animation is enacted when the screen pointer rolls-over a button to change the text label from 2-dimensional to 3-dimensional.

- Motion Tweening: For animations where an object is moved, rotated or transformed between two points. Motion tweens can also be used to fade colours from one shade to another, or to define 'alpha animations' where a colour or graphic is made transparent.
- Shape Tweening: For animations where a shape or graphic has a definite starting and end point but is distorted in intermediate frames to give the impression of 'moulding' from one shape to another.

The three types of animation that needed to be developed in order to fulfil the design were the frog, persona and piggy bank animations. Before animating the frames for these movies it was necessary to create appropriate graphics so that the start and end points could be set. Therefore appropriate bitmap images were taken from the internet and they were traced using the 'bitmap trace' function of flash as well as freehand tracing, using the mouse. Subsequent use of the graphical development tools made it possible to tailor the appearance of these graphics and their associated fill-colours to produce the required high-quality graphics that are appropriate for animations. Each animation was then created by converting an instance of each graphic to a movie clip, where the following tweens were specified on the timeline:

- Frog: (refer to figure 1, Implementation Appendix, section 3). Each frog animation was created using a number of motion tweens, both to animate the frog graphic between two points and to move the legs and arms of the frog whilst in motion.
- Persona: (refer to figure 2, Implementation Appendix, section 3). Each persona animation was created using layers; a graphics layer for the persona's head and an animation layer for the persona's mouth. The animation layer was self-authored using 6 different mouth graphics to create the different mouth shapes necessary to articulate words. A third layer for sound was defined where '.wav' sound files created in section 5.4.1 were attached to produce the required audible output.
  Interestingly, when sound was attached to the sound layer it was possible to see the peaks and troughs of the sound wave. Therefore it was possible to recognise points at which the mouth animation must be enacted (peaks in the sound wave) and when it must be still (troughs in the sound wave), which made the persona's articulation of words more accurate.
- Piggy Bank: An example of a piggy bank animation is the 'exploding pig' (refer to figure 3, Implementation Appendix, section 3). This was created using a number of self-authored frames to give the appearance that the pig is shaking. A shape tween was used to explode the pig, using an animation mask to ensure that the body of the pig was always visible. A motion tween was then used to perform an 'alpha animation' of the explosion to give the appearance that the broken pieces disappeared. A further motion tween was used to animate the pig running away from the scene.

To embed each animation in the scene of the movie so that it is played at the right time and also so that it is positioned in the right place, it is necessary to use placeholders. All instances of all animations required for a scene have to be placed *around* the stage (but out of view) so that they can be referenced using ActionScript coding. Each animation is given a placeholder, which is an invisible graphic that is placed in the appropriate position *on* the stage where the related animation needs to be placed. Its coordinates are then held in an array data structure and are referenced and applied to the appropriate movie clip at the time in the movie when the animation needs to be moved *onto* the stage and played.

After creating the required sounds, buttons, graphics and animations, and having stored these in the component library it was possible to create each flash file that comprises the system.

5.5 Creating Movies

When creating movies it is necessary to manipulate the movie playback in order to create the required interactivity. Frames are used in this context as they allow us to specify 'what should happen and when'. By naming particular frames it is possible to attach ActionScript and thus make the play-head jump to certain points of the movie timeline or execute functions at given times in order to support more advanced functionality (for example prepare a scene, play a movie clip or stop the movie play-head altogether). In summary, using frames allows us to dictate how the system performs.

In this section we will explain how the following flash files were implemented:

- `tutor_MAIN.swf`
- `tutor_teacher_controls.swf`
- `tutor_level_1.swf`

As stated in section 5.2 these files were the 'master copies' for producing all other files. Therefore by describing the implementation of each of these it is possible to represent the most important aspects of the implementation.

5.5.1 Main Menu

The main menu (`tutor_MAIN.swf`) is the screen that confronts the user when the system is launched. Accordingly it must initialise the settings for the flash player, as well as the settings for the main menu scene itself.

The frames that dictate the conditions for movie playback for the main menu are `init_MM` and `standard_MM`, which are played in the following order:

1. `Init_MM`: This frame enacts the flash `fscommand` to set the flash player to full screen. It also initialises the movie clip placeholder that is used as a reference for the coordinates of the interface persona animation. This frame is simply played through by the play-head.
2. `Standard_MM`: This frame uses the `stop()` function to stop the movie playback and wait user input. It positions the persona animation according to the coordinates of the placeholder and uses the `eval.(Persona).gotoAndPlay()` function to play the persona movie and thus output "choose a game". The frame also sets the key words to the 'output' dynamic text box[24] to state that the user must choose game.

Refer to Implementation Appendix section 4 for a definition of the layers that are related to this movie.

5.5.2 Teacher Controls

The teacher controls (`tutor_teacher_controls.swf`) is launched from the main menu, and contains a customisability option that allows the teacher to change the structure of the main menu to either 'temporal' or 'options'.

In contrast to the main menu, this movie contains 2 scenes; `teacher_controls` and `exit_confirmation`. Two scenes in the same movie essentially act as separate movie clips, although they are able to share components and layers. This means that we are able to reuse some of the layers of the `teacher_controls` scene in the `exit_confirmation` scene, which is important given that they have the same appearance.

[24] A dynamic text field is a text area that can be editing at any point in a movie to reflect text output. Importantly for the purposes of printing key words to screen, dynamic text can be set using ActionScript coding to relay appropriate information for example.

The frames of these scenes that dictate the conditions for movie playback are:

- `teacher_controls`: This scene contains 1 frame that is unlabelled. It enacts the `stop()` function to stop the movie when it is opened and thus wait for user input.
- `exit_confirmation`: This scene contains 1 frame that is unlabelled and has no attached ActionScript. It is not necessary to specify that the play-head should stop when this scene is opened as this is specified by the `gotoAndStop()` function enacted by clicking the 'exit system' button from the teacher controls scene. Hence on opening the movie is paused for user input.

Refer to Implementation Appendix section 4 for a definition of the layers that are related to both these scenes.

5.5.3 The Task

As already stated `tutor_level_1.swf` is the 'master copy' from which all other task levels are created. However it must be noted that there are small differences between the movie files for the levels of the task, given that they have different mathematical content and slightly different 'end of task' scenes. However, in order to make the implementation of these changes minimal and easy, `tutor_level_1.swf` was implemented in such a way that it was as modular as possible, making reuse, modification and maintenance simple.

An important problem to overcome when implementing the task was that flash does not have a built in `Sleep()` function similar to that of C or Java. Consequently, the only way to stop the movie for a prolonged period is to use the built-in `stop()` function and then get the user to start it again (using a button for example) to trigger the `play()` function. This is cumbersome and inflexible, and was not considered appropriate in certain parts of the interface. Therefore it was necessary to write a `Sleep()` function (see figure 5.3) that stops the movie and then starts it again automatically, after a given delay. This was used in the task to pause movie playback at certain points in the timeline, mainly so that the user had enough time to read and interpret information on the screen.

Figure 5.3: Source code for Sleep and Wake functions

```
//sto
funct

                                                             ached
}

//wake the movie back up!
function Wake() {
       clearInterval(timerID); //call this function only once
       play(); //start the movie again
}
```

The task was implemented in 4 'phases' each of which contained appropriate layers, frames and ActionScript to perform the required functionality. These phases are described in the following sections; refer to Implementation Appendix section 4 for a definition of layers associated with this movie and also the functions that were produced to support advanced functionality.

Initiation

When the task is initialised two frames are played to set up the scene and the mathematical information appropriate to the task level. This a key part of the task as all animations and interactivity depend on a set of properties and data structures being established before the movie play-head can continue along the timeline:

1. `Init task`: This frame calls the `InitPlaceholders()` function to initialise a set of arrays to hold the coordinates of all the placeholders for the animations that are used in the movie. It also calls the `InitQuestionArray()` function to initialise a further set of arrays to hold the information relevant to each question. Importantly this function initialises a 'questions' array that holds an identifier specific to each question of the task. This identifier is used to reference all the relevant animations and graphics, which is essential for 'setting up the scene' for each question posed. For example the components specific to question 7 are:
    - Question7 – instance name of the question text (positioned at the bottom of the screen)
    - PersonQu7 – instance name of the persona animation
    - Frog7 – instance name of appropriate frog animation

    The `InitQuestionArray()` function also initialises an 'answers array', which holds the correct answer for each question. This array is referenced for each question of the task as it is essential that we can tell with certainly what the *actual* answer is, given that the answer to one question is a red-herring to another.

2. `Start`: This frame sets up the global variables that are used as important flags and counters whilst the task is being performed. It calls the `QuestionOrder()` function to randomise the order that the questions are played in. This is a vitally important function of the task in accordance with functional requirement 3, which states that all tasks should 'change' to support the user's educational development and thus not simple repeat static information. The algorithm for randomising the question order is shown in figure 5.4.

Figure 5.4: Source code for randomising the question order for the task level

```
            umberOfQuestions; i++) {
        TempQArray[i] = i;
}
var QuestionsLeft = NumberOfQuestions; //counter for questions left
//randomise the question order
for (i= 0; i<=NumberOfQuestions; i++) {
        //generate a random number [0-9] for the question place
        var QuestionPlace = random(QuestionsLeft);
        //set this question in the final order
        GameQuestions[Count] = TempQArray[QuestionPlace];
        //if QuestionPlace is not the last in the sequence then
        //remove the used index from TempQArray and replace with the
        //number at the top of the scale
        if (QuestionPlace != QuestionsLeft) {
                TempQArray[QuestionPlace]=TempQArray[QuestionsLeft];
        }
```

Pose Questions

After initialisation the movie play-head falls into a loop where the user is posed 8 different questions to test their mathematical knowledge. In order to populate the summary sheet (which denotes the user's performance in each question of the task), it is necessary to record the question, the user's input and the answer in each iteration of the question loop. This is done by writing information to a dynamic text box called `print`, which is positioned in a frame labelled `#p` at the end of the timeline so that the user cannot see it or access it.

The loop is enacted by running through a series of 5 frames, where a 'next' button from the last frame takes the user back to the start of loop (where another question is setup and performed). By nesting the questions in a loop it is possible to increase the modularity of the system, and thus reduce redundancy as well as the programming effort. Moreover whilst *sat* in a loop it is possible to *wait* for user input when needed, meaning that the user is not constrained by a time limit when performing the task. The sequence of events for the loop is:

1. In the `play question` frame, the scene is prepared for the current question, and thus the appropriate persona movie, question text and frog animations are positioned on the stage (with reference to their placeholders) and the key words are set to the dynamic text box. Two variables are initialised; a counter for the number of try-agains (as the user is limited to attempting a question only twice), and a Boolean `waiting` flag, which is set to false to denote that we do not setup another question until this one is finished.
2. The play-head then falls into the `loop question` frame which simply sends the play-head back to the `play question` frame until the user enters an answer. This acts as a loop within the question loop, making it possible to wait for user input. The aforementioned procedure for setting up a new question (step 1) is not performed while `waiting` is set to false and thus we only break out of this loop when the user enters an answer. The functionality triggered by user input is detailed in figure 5.5.

Figure 5.5: Algorithm for assessing user input. This is performed when the user clicks on one of the lily pad buttons to enter an answer to the question.

```
function ButtonCondition(ButtonName, QuestionAnswer) {
                == answer) {
              //user correct - set summary sheet and answer feedback
              //go the answer response frame for a correct answer
              //award a gold coin
       }
       else if (name < answer) {
              //user wrong - set summary sheet and answer feedback
              //increment the number of tries
              //check if the user has had two tries
              if (NumberOfTries == 2) {
                     //log incorrect for summary sheet
                     //go the answer response frame for a wrong answer
              } else {
                     //answer is too few, user can try again
                     //set details of user input in summary sheet
              }
       }
       else if (name > answer) {
              //user wrong - increment the number of tries
              //check if the user has had two tries
              if (NumberOfTries == 2) {
                     //log incorrect for summary sheet
                     //go the answer response frame for a wrong answer
              } else {
                     //answer is too many, user can try again
                     //set details of user input in summary sheet
              }
       }
```

3. The `gotoAndPlay()` function is enacted in `ButtonCondition()` to move the play-head to the `answer response` frame after the user has answered the question correctly, or after they have had two try-agains. Before this though, the `user input` global variable is set to 'true' if the answer is correct or 'false' if wrong, and this Boolean is evaluated in the `answer response` frame to ensure that appropriate feedback is given:

- If the user gives a correct answer, the `CorrectAnswer()` function is enacted to play one of the three 'correct' persona movies, set appropriate key words to congratulate the user and update the summary sheet with this result.
- If the user gives a wrong answer, the `WrongAnswer()` function is enacted to play one of the two 'wrong' persona movies, set appropriate key words to commiserate the user and update the summary sheet with this result.

4. The play-head then moves the `demo ans` frame where the correct answer is demonstrated using the relevant frog animation and audible feedback. Consolidating children's learning by summarising the question through demonstration was stated as one of the key points of "good direct teaching" (DFEE, 1999), hence the importance of this frame. .
5. After the correct answer has been demonstrated the play-head moves to the `next qu` frame where playback is stopped (using the `stop()` function). In this frame we wait for the user to click the 'next' button in order to return to step 1 of the loop where the next question is setup and posed. If all questions have been played at this point the play-head does not move back to the start of the loop, instead it moves to the `reward scheme` frame, which is described in the next section.

Reward Scheme

After the 8th question has been completed the 'next' button enacts the `gotoAndStop()` function to move the play-head to the `reward scheme` frame instead of the `play question` frame. The `EndOfLevel()` function is called to calculate how many correct answers the user has given and this is written to both the summary sheet and the dynamic text box, providing feedback for the user. Subsequently the `SetUpRewardScheme()` function is used to prepare the scene with the appropriate persona movie and piggy bank animation. The `eval.(Piggy).gotoAndPlay()` function is then called to play the piggy bank animation and thus reward the user for their achievements. At this point we wait for the user to click the 'next' button to move to the `end of task` frame where they are able to choose what to do next. Closure (Dix et al, 2000) is provided by clicking this button, as it reaffirms that the user has finished the task.

End of Task

The `gotoAndStop()` function is enacted when the user clicks the 'next' button from the reward scheme and thus playback stops in this frame to wait for user input. The `SetUpNextStage()` function is used to prepare the scene with the appropriate persona movie and this is then played to feedbackback that the user must choose one of the end of the task options:

- Play the level again: `gotoAndPlay()` is used to return to the start of the task.
- Move to the next level: `loadMovie()` is called to load the next level of the task.
- Go back to the main menu: `loadMovie()` is called to load the main menu.
- Print the task summary sheet: Because information has been written to the summary sheet throughout the task we only need to call the `printNum(0, "bframe")` function to print a sumarry. Importantly, the frame which holds the print dynamic text box (containing the summary information) is labelled `#p` which tells the print function to *only* print this frame and not all frames in the movie.
  It is important to note that when implementing this function, it was apparent that the support flash offers for printing frames of a movie is not sufficient enough to implement the design solution specifying that the print function should not 'pop-up' a confirmation window.

After implementation of all movies that were specified in section 5.2, publication was required to convert the '.fla' flash files into movies that were suitable for testing.

5.6 Publication

Published flash movies are commonly '.swf' shockwave files. However, shockwave files cannot play on all computer systems without an appropriate media player, or html flash-plug-in, which of course is inappropriate given non-functional requirement 19. Accordingly 'tutor_MAIN.swf' (the main menu for the temporal structure) was published in '.exe' format. When published in this format the movie file is packaged with its own flash player, and because the main menu is the 'starting point' for the system, all '.swf' movie files loaded from the main menu, or any subsequent scene, are loaded in the packaged player.

For the purposes of testing and evaluation this means that `tutor_MAIN.exe` and other '.swf' movie files could be written to CD and then run consistently and predictably in the classroom at full screen resolution. To ease the teacher's role for running the software during evaluation, an `autorun.inf`[25] file was created and written to the CD with the movie files. When inserted into the CD-drive of any computer running a Windows operating system an `autorun.inf` is searched for, and by setting tutor_MAIN.exe as the file to autorun (refer to figure 5.6) it was possible to ensure that the system initialised the main menu every time it is used.

Figure 5.6: Source code for autorun.inf

```
[autorun]
```

Refer to Appendix section 5 for the interface maps of the both the temporal and options structures. These diagrams contain screen shots of the interface and show how each of the screens interact.

5.7 Testing

System testing is an essential stage in the development of any computer system as it ensures that any problems or errors produced by the system are recognised and rectified. As specified in non-functional requirement 17 performing system testing is essential so that the system is reliable and thus does not produce any unexpected behaviour. Equally though, the overall aim of the project is to produce a prototype solution that is suitably functional for the purposes of deployment and testing in the classroom. Accordingly the system must be usable for research purposes and therefore all relevant functions must be tested to ensure they run without fault. This does not necessarily comply with the exhaustive techniques for safety-critical systems, but in this context this is not necessary as the tests performed were adequate and thorough enough given the project aim.

By implementing the system in a series of incremental steps as detailed in this chapter, it was possible to apply unit testing on each completed section to verify that it met the requirements as well as the desired functionality. This meant that any immediate problems could be rectified before they could become nested in a hierarchy of errors. After implementation of the prototype was completed, the whole system was tested using black box testing techniques (refer to the Implementation appendix, section 6 for a sample of these tests) to ensure that it functioned without error and thus was suitable for the aforementioned purposes of evaluation. Of course any errors or unexpected behaviour that were identified during testing were corrected and the code section was then re-tested to ensure the problem was fixed.

[25] autorun.inf files are only applicable to windows systems. However, given the computing specification of the classroom machines in non-functional requirement 18, this is clearly suitable.

6 Evaluation

6.1 Introduction

Evaluation of the tutoring system is an essential part of the project as it allows us to assess whether the solution meets the key objectives as well as the system requirements. In this context it was necessary to examine whether the tutor supported and aided the development of mathematics education and also whether it was truly usable in the context of the foundation stage classroom.

In order to evaluate both the usability and learning benefit of the tutor, consideration was given to the methods that would provide the basis for effective evaluation. In terms of usability, the data gathering techniques of naturalistic observation (Preece et al, 2002) as well as questioning[26] were thought to be most relevant for eliciting information *whilst* the children used the tutoring system. This meant that emerging issues could be noted and discussed as they happened. In consideration of the problems encountered at the design stage regarding the children's limited ability to articulate their thoughts with clear meaning, it was decided that these techniques would remove the need for children to perform evaluative thought which would benefit the results gathered.

In terms of usability it was decided that two methods for evaluation should be performed where two different test controls are manipulated in order to gather a breadth of results. Clearly this provides the basis for more substantiated conclusions could be drawn. The two evaluations chosen were:

- The performance of children who had used the system, against those that had not, and
- The performance of the children at the start of the week, against the performance at the end of the week. Importantly, it was decided that only half of the children in the class should be allowed to use the system[27] so that at the end of the week a subjective comparison could be drawn and any notable differences could be examined.

The tests performed that allowed us to perform the evaluations over usability and learning prowess were performed in Louise Aylott's foundation stage classroom at North Baddesley Infant School, Southampton, between 19th April 2004 and 23rd April 2004. The system was run on one of the three computers that are stationed in the classroom and was used by the children all week.

6.2 Usability Evaluation

As already stated, observation and questioning were used over the course of week to perform the required assessment and gather appropriate information. Important issues were noted at the start of the week and then reinvestigated at the end of the week to examine whether it was possible to identify any particularly good aspects of the system that clearly fulfilled the system requirements, or any particularly bad aspects that would require redesign in the future.

[26] Questioning was directed at children when they were using the system to elicit information about their thought processes, opinions and expectations from using the system.
[27] The children were divided into a 'testing' and 'non-testing' group such that both groups had a relatively even ability level. The division was performed under the direction of the teacher.

6.2.1 Monday

Tests carried out on Monday 19th April, 2004 yielded the following results.

Given the instruction; “choose a game” from the main menu, all the children correctly referenced the frog in the pond game in the house of games to initiate the task, which infers that they had a good initial understanding of the navigational structure. Notably none of the children enquired about the purpose of the ‘teacher controls’ button. On questioning it emerged that the children either did not think it was relevant to them or ‘did not notice’ it existed as it was separate from the house of games. This complies with the aim of the design as clearly the children are not meant to access the teacher controls.

As was apparent in the main menu, the children listened to the persona’s voice as the primary means for interpreting each question of the task, where the question text and key words were mainly used as ‘fall backs’. The children recognised the persona as being the authority figure of the interface, which was reinforced through the statements; “…he tells me what to do” and “… I have to listen to what he says”. In relation to this though, it was very rare that the children used the repeat button to repeat these instructions. On questioning it was clear that the children did not understand the purpose of this button, which implies they did not understand the button label ‘repeat’ as it relates to the persona’s voice; clearly they could not make the relation ‘I did not hear the voice, so I’ll click the button’. It is not easy to suggest a design solution to this problem. The Tizzy’s Toybox tutor that was evaluated in the requirements analysis used a ‘lips’ graphic to denote ‘repeat’, which incurred similar misunderstanding. Accordingly a design solution might be to remove the button altogether and have the persona repeat the instructions in a loop. Therefore if the child does not input an answer after 20 seconds (for example) the instruction or feedback could be repeated without the need for the user to prompt it. However, this solution would need to be tested, as repeating the audible instructions could interrupt the child whilst they are calculating their answer or become annoying for the user (or indeed the rest of the classroom), which were concerns identified in the design stage.

When performing the task at all levels, it was noticeable that the children’s control over the mouse was far ranging, from some who could not accurately control the screen pointer, to others who did not have appropriate motor skills to consistently hit the left mouse button. Given this constraint, the ‘roll over’ and ‘hit’ effects for the buttons were advantageous and were relied upon as a method for the child to assess whether they were on the button hot-spot or whether they had clicked a button.

A second observation noted when the children were performing the task was that some of them repeated the language communicated through the interface. An example being: “…I think 1 more than 4 is 5”. Of course by repeating this audible communication the children are able to practice using mathematical language in appropriate contexts, which was specified as an important objective of mathematics teaching by the foundation stage teachers in the requirements analysis chapter.

A further positive observation noted when the children were performing the task, was their acceptance and enjoyment of interacting with the learning metaphor. The children closely followed the movements of the frog, and on questioning it emerged that they understand the representation of the pond. Moreover the statement; “he jumps to the right one every time” would imply that they also interpreted the purpose of the animation accurately and benefited from it.

Through questioning it was possible to identify the importance of the question text at the bottom of the screen for helping the children to interpret the purpose of each question. This was particularly salient when performing level 3 of the task, where in order to solve the problem “3 plus 3” the children referenced the question text at the bottom of the screen and counted 3 on each hand before counting all their fingers to calculate their answer. The question text is a beneficial design solution as it provides a 'permanent' reference for the key numbers for the question which must be reference in order to calculate an answer.

The children were confused by the 'next' button that appears at the end of each question. They did not realise that this could be used to move onto the next question, and thus they misinterpreted the text and arrow label. Many children asked "what do I do now" or made the mistake of thinking that the replay of the answer was the start of another question. This is a usability issue encountered for first-time users. A possible solution would be to have a pause at the end of the question instead of having a 'next' button, so that the next question is posed after 5 seconds (for example). This would mean that the child does not have to click a button to move to the next question.

The children easily interpreted the purpose of the progress indicator and had a well developed understanding of its relevance to the task. On questioning all the children could recognise that they received a coin for each correct answer, and they strived to gain as many as they could in the task; "I'm going to fill all the gaps…" and "…I've got 4 right so far". The were able to use the progress indicator to adjudge the number of gold coins they had accumulated, and the number of questions they had left until the end of the task. Moreover, the gold coins formed a means for comparison between the children, highlighted by; "you're doing well" and "I got more right than you."
In relation to this, the piggy bank reward scheme was also a success. Many of the children laughed and gasped when the pig exploded, and accurately made the relation that the pig exploding denoted a higher level of achievement than making it jump up and down. They strived to make the pig crack which infers that they were strongly motivated to get as many questions correct as possible in the task. This was reinforced through the comments "I made him crack!" and "I don't think I have got enough to make the pig crack now." These results were particularly pleasing as the reward scheme and progress indicator were identified as important design solutions to the requirements for providing a goal for learning as well as providing visibility of system status (Nielsen, 2001).

At the end of the task, the children were able to identify that they could print their work, with reference to the print button. The print confirmation window did cause a usability problem however, as was expected after its implementation. The children had to be instructed to click 'ok' from this window which of course does comply with the required intuitive nature specified for the system. However, it was apparent that after being given this instruction the children had no subsequent problems when printing. Moreover, it was noticeable that a hardcopy printout of their work provided some satisfaction for the children, who were very keen to show the teacher what they had done. This relates to Keller (1983) who stated that a key aspect of motivation is to satisfy the user.

A notable usability issue was identified at the end of the task. The children could not interpret the meaning of the buttons for repeating the current level or navigating to the next level, and thus many simply clicked the button 'home' as they did not know what else to do. Accordingly we can recognise that these buttons are not 'intuitive enough' for the children and it is a major usability issue for them to have to *learn* what each of these buttons do and then recall this knowledge every time they perform the task. An interim solution was to use the 'options' structure where the children were able to follow their natural train of thought - to navigate back to the main menu and then initiate another level of the task. Using this structure triggered an almost instant beneficial reaction from the children.

6.2.2 Friday

Tests carried out on Friday 23rd April, 2004 yielded the following results. It is important to note that the children continued to use the options structure for these tests.

After a week of use the children could navigate the interface with much greater ease and control. All the children could initiate a learning session by clicking on a button to one of the task levels and on questioning it emerged that they could also associate each level with its related level of difficulty, such that game 3 was more difficult than game 1. This was reinforced by the statements; "…just the difficult one to go now" and "…this one is more difficult." These observations would suggest that the options structure is far more suitable for use in the classroom than the temporal structure, as it is clearly associated with a smaller learning effort.

As was apparent on Monday, the button effects were of importance to the children, who relied on the 'roll-over' effect in particular to assess whether they were 'on' the button hotspot. Accordingly we can conclude that they were an essential design solution to the constraint of children having relatively poor control over the mouse.

The usability issue identified on Monday concerning the 'next' button for moving to the question in the task was not noted as a problem on Friday as all the children would quickly navigate to the 'next' button after answering a question. Therefore it is questionable whether the proposed design solution to have a pause at the end of a question should be upheld. Further testing with novice users would be required to examine whether the 'next' button is too significant a usability problem to overcome when using the system for the first time, whereby the proposed redesign would yield a more intuitive task structure.

On Friday the number of children that repeated the persona's mathematical language was more common than that observed on Monday. This was shown by the statements; "what is 1 more than 4 is 5…that's 6" for example. As identified in the literature review, Hitcham (2000) stated the importance of children developing their communication skills and ability to use mathematical language correctly, which was also stated as an important objective of the national curriculum. The tutoring system supports this, and encourages children to practice it.

It was noted that more children used the 'home' button to break their learning session than at the start of the week. If the allotted lesson time ended in the middle of their learning session the children would almost immediately navigate to the 'home' button in order to quit the task. This is a very favourable result as it means that Nielsen's (2001) usability guideline of providing an emergency exit is fulfilled and is well understood by the children. However it was noted that very few children would click the home button without being prompted by the end of the lesson. They evidently wanted to complete each level and would only quit when they got to the end – after the reward scheme.

At the end of the task, the children readily printed their task summaries and thus overcame the problem of the print confirmation window. They were notably enthusiastic to show the teacher what they achieved, although through questioning it was evident that few children could actually identify what this summary represented. However, Louise Aylott stated that she was pleased with the format of the summaries as she could easily interpret the child's achievements and subsequently add it directly to the child's foundation stage profile. This indicates that the tutor integrated well into classroom practice and school procedure, and thus would be a suitable source for adding work to each child's foundation stage profile.

After printing their work, all children clicked the 'home' button to return to the main menu. Importantly, none of the children clicked the 'repeat current game' button at the end of the task, preferring to return to the main menu and launch another level of the task. As was identified on Monday, this button is redundant to the system as the children could not overcome the initial problem they had with interpreting its purpose.

6.2.3 Teacher Observations

Through discussion with Louise Aylott about how the system performed when it was used throughout the week, two key points emerged:

Louise expressed a clear preference for the 'options' structure. The reasoning was that she thought it was more appropriate for the children to be able to access any level of the game from the main menu so that they could break their learning session (at level 2 for example) and then return to the same level later. This preference was largely dictated by the nature of the school day, where break-time, lunchtime and other classroom lessons make it difficult for a child to be undisturbed for prolonged periods.

By using the options structure the problem noted on Monday where the children were unable to interpret the meaning of the buttons at the end of each level[28] was largely eliminated. Louise stated that children could more easily make the inference that to start a new level they should return to the main menu and choose a different button, as this meant that they initiated each level of the task from a central place. Accordingly it is possible to note the importance of having a 'central place' as a guideline for creating tutoring systems, due to the benefit it offers for providing a familiar and intuitive navigational structure.

From this we can surmise that it might not be appropriate to have a temporal task structure, where a sequence of difficulty levels is dictated. This finding goes against Hammond (2000) whose theory of 'scaffolding' dictates that children should be exposed to concepts whereby their understanding is expanded through exposure to more difficult applications that build on previous knowledge. This was identified as a very important theory in the literature review and thus it would be wrong to dismiss it. Accordingly it would be appropriate to combine our findings with that of Hammond, and perform a redesign of the 'options' structure. By redesigning the main menu to 'recommend' the next level that the user should attempt by highlighting it, or by using audible instruction, we may be able to persuade the child to perform the task in sequence, whilst still providing the freedom for them to initiate any level. This would more accurately suit the activities of the classroom as it would allow the children to return to any level of the task if they had to break their learning session early.

A second key observation was that the children were very enthusiastic to use the system. This was illustrated by one of the children, who drew a picture of the system's interface in her spare time, but it was also apparent from the widespread discussion of the learning metaphor, and the reward scheme in the classroom. Louise Aylott stated that the children had a strong desire to gather as many gold coins as possible so that the final animation of "making the pig crack" was played. Louise said that the reward scheme provided a notable means for comparison and competition between the children, who used the computer more frequently in order to achieve this result. Of course through more frequent use the children became more proficient at performing the task (see section 7.3) which in turn will improve their numeracy skills – fulfilling a key objective for the project.

6.2.4 Conclusions

The results gathered from both the children and the teacher were very positive for the most part, and they denote a notable success for many of the design solutions to the requirements.

From the evaluation we have identified some issues, for which the suggested redesigns would refine the system's usability. Future work would involve implementing these redesigns, although none of the problems identified were so problematic that they impacted the children's ability to use the system effectively or pick-up how to perform their tasks. This indicates that the requirements for making the system intuitive and easy to understand have been met.

[28] Buttons for repeating the level or navigate to the next level

Certainly after using the system for a week the children were very proficient users in that they could quickly and easily navigate the interface, denoting a detailed understanding of the actions performed by clicking particular buttons. They also demonstrated a liking for the learning metaphor, and seemed to interpret the purpose of the frog animation and lily pad buttons with considerable ease. Moreover the enthusiasm shown to use the system and complete the task levels in order to achieve feedback and rewards, would indicate that the system improves upon the tutors that were evaluated in the requirements analysis on many counts. When observing the children using the evaluated systems they were much slower to perform actions at the interface as they were unsure of what would happen. Also the relatively weak learning metaphors and lack of local goals for completing their tasks meant that the children tired of using the systems, and the enthusiasm for interaction was notably lower that that observed during the evaluation of 'frog in the pond'.

A more general point that emerged from evaluating was that when confronted with a new software product the children are not always confident that they can pick up how to use it, simply because it is unfamiliar. This was reinforced by two statements made by the children on Monday; "I haven't used this before, how do I…?" and "can I click this button?" From this we can recognise that the children attach a stigma to performing new activities that they have not encountered before, even if they are able to make inferences about how to apply their knowledge at the system interface. Through questioning it emerged that the reason for this is that they do not want to make mistakes. However, as was apparent in the comparison between Monday and Friday, the children rapidly overcame this when using the tutoring system, and thus we can conclude that the system has a suitably short learning time for novice users.

6.3 Learning Evaluation

In addition to the naturalistic observation of system usability performed on Monday and Friday, the two formal tests described in section 6.1 were performed over the course of the week to assess the learning quality of the tutoring system:

1. Testing evaluated using the tutor: A representative set of children from the 'testing' group were chosen and their progress was monitored closely throughout the week. With the help of the teacher the sample of children chosen for this were two with relatively low ability, two with mid-range ability and two who were high achievers. These children were asked to perform all levels of the task everyday of the week, and also to print a summary of each level on each day, so that it was possible to gather useful data which provides the means for comparing their progress.
2. Testing led by the teacher: Louise Aylott performed a lesson of 'number in the bag' with the children on the Monday morning to assess their understanding of the concepts; "1 more than", "2 more than" and simple addition, before they had used the tutoring system. She then repeated this lesson on Friday after the 'tested' group of children had used the system all week. Louise suggested that this lesson was suitable because it is fundamental to foundation stage mathematics leaning and it was very closely related to the concepts that children could practice using the tutoring system, and thus she would be able to evaluate whether any improvements were made.

   Through discussion on Friday it was possible to assess the changes in behaviour and understanding that she had noted between the two lessons, and subsequently draw conclusions about any advancements that the tested group had made over the non-tested group.

6.3.1 Test 1 – Tests Using the Tutoring System

Refer to Evaluation Appendix section 1 for a summary of the results gained for the tests that were performed.

Monday Testing

Tests carried out on Monday 19th April, 2004 to assess the systems prowess at supporting learning when performing all 8 questions for each level of the task, yielded the following results:

All the children were very good at level 1, due to its very close relation to the activities of the classroom. This was clear from the speed at which they calculated their answers. Questioning revealed that the children performed the calculation in their head, without the need for using their fingers or the lily pads as an application to aid them, which was reflected in the results gathered; all the children scored maximum marks, apart from participant 3 who scored 6. There was also very few 'try agains' for this level (an average of 1.6 per child), indicating that they very quickly picked up how to interpret the instructions and input their answer.

Some children struggled to adjust to the different challenge offered by level 2, when performing it directly after level 1. As was apparent in the level 2 summaries, the children applied their knowledge from the previous level and made the incorrect inference that they should calculate '1 more' than the starting number, instead of '2 more', hence the large number of try-agains[29], thus relating to Bruner et al's (1971) theory of overextension of knowledge. It was noticeable for some children that it took 3 questions or more for them to correct this mistake before they took hindrance of the instructions and feedback being given. From this we can note that children gave the question a lack of attention as they were comfortable with performing the questions of the previous level and thus carried on without due care for detail. However, the children still got an average of 5.8 questions correct for the level, which would indicate that were able to apply their knowledge reasonably successfully, given that this level is of course harder than level 1.

Through questioning it emerged that the children used a similar technique to level 1 for calculating their answers, in that they performed the calculation in their head. However, it was also noted that three of the children pointed at the screen and counted the lily pads, which would indicate that they provide a useful application for the children to calculate their answer.

The children found level 3 a very different challenge to level 2. The results gathered show that they got an average of 5.8 questions right, but required fewer 'try agains' than in the previous level. The reasoning for this was that the children paid more attention to the questions that were posed and also they used their fingers as an application for calculating their answers, which is consistent with the method taught and applied in the classroom. This was in contrast to the previous level, where they tried to perform the calculation in their head. Importantly it was noted that some children were slow to make the inference that they needed to use their fingers in order to calculate their answer, and two children could only do this after instruction from the teacher. This would imply that they did not realise that the question required them to perform a calculation over addition and therefore a possible extension to the tutor may be to suggest that the children should use their fingers as a strategy for calculation. However when they did use their fingers as a concrete reference for performing the sum there were clear benefits in the results gained. This would imply that level 3 provides an application for practicing sums of addition using classroom taught techniques for calculation.

When performing level 3 it was noticeable that the children found the questions requiring them to calculate +4 or +5 to be very difficult, and we can recognise that this was perhaps due to the

[29] An average of 5.5 try agains was observed for each child.

fact that the 'jump' in difficulty from the previous level was too high. A possible design solution to this would be to have two different levels for addition. In a similar way to the relationship between level 1 and level 2 we could have an 'easier' level with sums of +1, +2 and +3 for example (which the children were more successful at) followed by a harder level with more difficult sums for the children to expand their knowledge within.

When performing the task at all levels it was clear that only some of the children were able to apply the inference that if their first answer was wrong by being "too few" for example, they should input an answer greater than their previous one. The children were able to apply this inference with greater success if they input 8 for example and the answer was too few (in which case 9 and 10 are the only options). It was more common that the children would try to totally recalculate their answer with reference to the question text at the bottom of the screen. Of course this is a favourable outcome, as it indicates that they recognised their mistake and took steps to reform it, from a familiar starting point. This was reinforced by the results taken for level two in particular. On Monday the children used an average of 5.5 'try agains' for the 8 questions indicating that they got 5.5 questions wrong at the first attempt. However the average number of questions right was 5.8, meaning that it was common for the children to recalculate their answer correctly and thus attempt to reform their initial mistake.

Friday Testing

Tests carried out on Friday 23rd April, 2004 to assess the systems prowess at supporting learning when performing all 8 questions for each level of the task, yielded the following results:

The results for Level 1 show that all children answered all 8 questions correctly. Also there was a very small average of 0.5 'try agains' per child which indicates an improvement over Monday's results. From this we can infer that the children of all ability levels have become more competent at performing simple questions concerning the number sequence. From observation it was also noted that the children were very quick at calculating their answers, and many seemed to produce 'automatic' responses; "1 more than 6; that's 7!" This relates to Bruner (1967) who stated that accelerated development is enacted through 'extra teaching' similar to that offered by the tutoring system. He stated that allowing children to practice applying their knowledge where they are given appropriate feedback, allows them to reach a point where they are comfortable and confident at performing questions relating to some mathematical concept. Accordingly we can infer that the tutoring system may aid accelerated development, although the testing and evaluation to support this with more certainty would need to be performed in more depth.

Level 2 was also performed with more ease than on Monday, and a significant improvement was noted in the results. That was particularly apparent with participant 3 who improved from 3 questions right on Monday to 6 questions right on Friday, and participant 6 who improved from 4 questions right on Monday to 6 questions right on Friday. Moreover less 'try agains' were required (an average of 2.8 per child, compared with 5.5 on Monday) which would indicate that the problems the children had on Monday with adjusting between levels 1 and 2 were improved upon. This was upheld through observation, where it emerged that the children had a good understanding of the difference between the tasks.

When observing the children performing level 3, it was apparent that they all relied on the application of using their fingers to calculate their answers. This is consistent with the observation noted on Monday, which concluded that children use the system as a tool for practicing addition, using classroom methods, instead of using the lily pads as a computer-based application. This would add weight to the previous argument for having the 'fingers application' on the screen to prompt the children to use it as a strategy for calculation. In accordance with this observation, the results do not show the same level of improvement as that seen in levels 1 and 2, where the children clearly developed their understanding as a result of applying their knowledge *within* the context of the tutoring system. However it was noted that the children answered 6.8 questions right on Friday compared with 5.8 on Monday and they also required slightly fewer 'try agains' which would suggest they can more accurately use the 'fingers application' to achieve accurate results. Therefore we cannot discount the educational benefit of this level for aiding the children's understanding of addition.

A second observation that emerged from analysing the summary sheets was that the children got the +4 and +5 sums wrong more frequently than the other questions, which is consistent with the start of the week. Accordingly we can conclude that it would be appropriate to redesign this level into 2 levels ('easy' and 'hard'), as was previously suggested. This would comply with Hammond's (2000) theory of scaffolding, and would possibly lead to similar educational benefits as those observed with levels 1 and 2.

6.3.2 Test 2 – Tests Led by the Teacher

As already stated, Louise Aylott performed the 'number in the bag' lesson at the start of the week with the whole class, and took special care to observe the children's responses for the questions '1 more than' or '2 more than' a given starting number, as well as other questions over addition. At the end of the week Louise repeated this lesson and through discussion the following observations emerged:

The 'tested' children (i.e. those that had used the computers) were noted as being quicker than some of the other children in the classroom when responding to all questions '1 more than' a starting number. The majority of the children in the classroom (tested or otherwise) could respond to these questions almost automatically, due to the frequency at which this concept is practiced in the classroom. However, Louise did state that the lower ability children that were tested were more responsive than at the start of the week, and seemed more confident when giving an answer. Although this evidence is subjective, it concurs with some of the results from tests on the tutor, and thus we can conclude that by gaining an exposure to similar questions when using the tutoring system their performance during the classroom test was benefited.

When testing the children on numbers that are '2 more than' a starting number, more notable differences were observed than in level 1. Louise stated that some of the non-tested children (i.e. those that hadn't used the tutoring system) were quite slow to pick up how to answer the questions, and initially many mistakes were made. This draws parallels with the tested children when they performed level 2 of the task during Monday's testing session. However, Louise stated her surprise at the speed at which the tested children adapted to the more difficult questions, and offered correct answers. This is as a result of their exposure to similar questions when using the system, which denotes a significant learning quality offered by the tutor.

Louise stated that when she posed a set of questions to test the children's understanding of addition she did not see remarkable differences between the two groups of children. The 'tested' group were quicker to make the inference that they needed to use their fingers as an application to calculate their answer as they were more familiar with this. However, once the other children did the same, Louise stated that the number of correct answers received was evenly spread. From this we can note that the system provided a beneficial learning exercise, but clearly it needs to be extended to teach, reinforce and allow practice for applying the counting fingers strategy. As was seen during testing on Monday, many of the children were simply unable to even start answering a question as they could not make the relation between the mathematical problem and the application of using their fingers. Due to the tested children's familiarity with this application they were able to make this inference quickly and without the prompt of the teacher, which clearly denotes an advancement in their knowledge.

6.3.3 Conclusions Drawn from Learning Evaluation

As was demonstrated in the usability tests, significant advancements were seen in the results at the start of the week when compared with the end of the week. In the usability tests this indicated that the system became more familiar and was suitable for supporting the user's tasks, but in the context of a learning evaluation this indicates that the system has added value to each child's understanding of number, the number sequence and addition. Investigating whether this could be achieved was cited as one of the key objectives for the project and thus we can recognise that it has been met, at some level.

With reference to the Evaluation Appendix section 2, we can see that all the children on all levels of the task showed an improvement over the course of the week, or at least equalled their performance. However the observations noted during testing revealed that the advancements seen in levels 1 and 2 were more significant than that of level 3. This was because levels 1 and 2 provided the means for children to genuinely practice applying their knowledge in the context of the learning metaphor where appropriate and relevant mathematical language aided their understanding. This gave the children an advantage during the classroom lesson within which the teacher was able to evaluate their improvement.

Level 3 provided the application for the children to practice questions of addition, although the system itself did not appear to offer any advantages for accelerating development. A possible reason for this was the rather sharp 'jump' in difficulty from level 2 to 3, as the theory of scaffolding (Hammond, 2000) was not necessary complied with. However, it was still noted that the tested children were more aware of the fact that to calculate questions over addition they should use the 'fingers application', which would denote some improvement in the connections they have made with their existing knowledge.

6.4 Conclusion to Evaluation

The results gathered have allowed many interesting conclusions to be drawn about both the usability and learning prowess of the system, as well as the approaches taken.

In consideration of the usability evaluation we were able to note some issues that caused initial confusion for the children. Observation was a useful technique for noticing these issues, and when combined with questioning it was possible to draw out the particular problems that the children had. Similar advantages were also noted for observing good aspects of the system. When the participants smiled and laughed it was clear that the tutor provided some level of satisfaction, and through observing the children counting on their fingers or repeating the mathematical language we could also note that the questions posed a challenge and genuinely required the children to *think*. In accordance with this, observation and questioning can be cited as very suitable complementary techniques for performing evaluations concerning children.

Before performing the learning evaluation it was stated that by nature the views of the teacher were subjective and could not be solely relied upon for drawing substantiated conclusions. However, when combined with observation and questioning many of the issues that were raised were reinforced and affirmed, which provided evidence that the tutor was successful in accelerating the development of the tested children in the time period used. In reflection of the approach taken we can recognise that a more in-depth evaluation might be necessary to include comparing the tutor's performance against that of other ICT media, paper-based tutoring sheets or books, in order to see whether the particular style or format offered by the tutor for learning is beneficial, or whether novelty appeal was contributory to the results. However, it is important to consider that these tutoring media should be complementary to each other, and so stating that one method is better than another, is not necessary a true reflection on the educational value they offer.

7 Conclusions

In the Introduction to the project the scope for using computer-based tuition in foundation stage education was identified, and it was stated that computers are a possible solution to the problem of providing individual tuition for mathematics. In this context tutoring software was described as being useful only as a supplementary teaching tool because the concept of designing for children in a mathematical domain is not necessarily understood in enough depth to 'create successful learning tools' that could be solely relied upon to aid education development. With this problem being the driving force for the project, the aims and objectives that were set out have formed a specification for the work undertaken in the preceding chapters.

A difficult problem to overcome in the early part of the project was to set a direction, given the enormous scope of approaches that could be taken for meeting the original objectives. The number of mathematical concepts that could be supported, the number of methods for presenting information in an appropriate educational context, and the number of curriculum goals that could be supported formed a vast spectrum. However, the age of the target user group coupled with the need to investigate how a computer could be used to aid mathematics teaching in a way that utilised its potential to provide individual tuition formed ever-present constraints on the system design. By identifying the core issues and by collaborating with a variety of system stakeholders it was possible to form an appropriately scoped approach, such that the educational material presented and the mode of child-computer interaction was relevant given the capabilities of foundation stage children.

The literature review provided a largely theoretical grounding in the main issues relevant to the problem, and it provided the justification for using a computer-based tutor for educational purposes. It included an investigation into relevant HCI issues, which has formed an important underpinning to the project with the aim of satisfying the objective of making the system highly usable, easy to learn and relevant for foundation stage children. As a whole the literature review helped to direct thought into the pertinent issues related to the problem such that it was possible to begin identifying aspects of the system that could support educational development. However, the theoretical understanding borne out of this research was not substantial enough to make accurate assumptions about the users or the activities they perform. Practical sources were sought and evaluated through empirical studies and iterative design allowed a more complete set of system requirements to be compiled. Investigation and observation of teaching and learning processes in their natural environment coupled with information gained through discussion with appropriate system stakeholders allowed us to deepen our understanding and thus identify the issues grounded in both theory and practice. By evaluating existing tutoring systems at this stage we were also able 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. Only by combining the knowledge gained from theoretical and practical research was it possible to fulfil the objective of gaining an in-depth understanding of the problem domain.

The implementation of the prototype solution was the overall aim of the project. It dealt with the issue of effectively supporting educational development by providing carefully planned opportunities for practice, and through evaluation of the system's usability and learning prowess we are able to conclude that it was successful at doing this. An overriding reason for this success was the superb support given by Macromedia Flash for developing graphics intensive applications, which enabled the design solutions to be implemented with accuracy. Particularly the representation of the learning metaphor enabled children to more easily relate to the underlying mathematical concepts and thus create associations between them and the real-world, which Vygotsky, Donaldson (1978) and Liebeck (1984) stated as essential to learning.

The importance of achieving this was stated in the literature review, where Piaget's research indicated that children in the "intuitive period" solved problems by empirical testing rather than logical reasoning and therefore there was an inherent need for appropriate mathematical problems to be posed so that the children could apply similar methods to those used in the classroom for calculating their answers.

Through all research activities it emerged that a key requirement for the system was that it needed to be easily and naturally incorporated into foundation stage lesson planning in a way that was educationally valuable. By nature this involved producing a highly intuitive system where the computer is transparent; enabling the children to apply their knowledge without constraint. Clearly then relevant HCI issues governing usability (Nielsen, 2000) and in particular learnability (Dix et al, 2000) were of vital importance so that the effort expended to be in a position to learn was as little as possible and the demands on the children for *recalling how to use* the system were minimal. In summary, complying with these guidelines provides the basis from which an educationally valuable system can be built. However, to simply apply these principles is not necessarily sufficient when children are the target users for a computer system. Having completed this project it has emerged that HCI issues and usability principles need to be considered with reference to theoretical research concerning human memory and cognitive structures, so that the system is not only intuitive but also appropriate given the thought processes, cognitive abilities and limitations characteristic of young children.

Accordingly communicating appropriate, interpretable information through the interface was important to making the tutor easily understandable, so that children had access to all the information they required. Audible communication was used as the primary means for doing this, where appropriate language was used to mirror the mode of communication in the classroom. In using text key words emerged during the requirements analysis as a method for providing a reference for the children that was useful for them when calculating their answers. Through questioning during evaluation it was noted that children used a combination of these communicative media to interpret instructions and feedback, and thus it is possible to recognise they were both essential to the tutor's success.

In consideration of the underdeveloped motor skills of children, minimising error at the interface was also noted as being vital to the success of the system, and accordingly it can be cited as a necessary consideration for any child-oriented computer system. Through wide-spacing, buffer areas and sensitive button effects, children were able to accurately map their desires to computer input and interpret what they were doing with greater accuracy, control, and fewer mistakes, which Newton (2001) stated as being essential if "learners are to benefit from a task..." This relates to the previous discussion regarding the relevance of HCI issues, but its importance is clear because it allows the children to access the functionality of the system with a distinct level of ease.

Through research activities we identified the need for children to have local goals for completing tasks so that they are motivated to complete their learning session, for which Keller (1983) provided a number of guidelines. Where children are concerned, it is clear that the motivation for learning is not directly fuelled by the desire to develop their knowledge or get a high percentage of questions correct. Instead visual rewards that can be achieved in the short-term are favoured. Therefore it was important that the system used a reward scheme to reward the children on a per-question and per-task basis in ways that they appreciated and could relate to. When evaluating the prototype solution this point was reinforced. The children's desire to enact the reward scheme animation and also to print a summary of their performance in the task had an enormous effect on their desire to repeat it with a view to gaining better results. This had the effect of increasing their exposure to the mathematical concepts which was shown to accelerate their development even over a relatively short period of time.

In order to make the system relevant for the purposes of teaching given the experiences of the children and the objectives of the national curriculum we can reference the sentiments of "good, direct teaching" (DFEE, 1999) from the literature review. This provided guidance on how the effective assimilation of knowledge is achieved, and it is possible to recognise that the tutoring system follows these principles very closely. The pattern of instructing, questioning and consolidating is enacted in each question posed by the system, which is stated as being the "ideal method for teaching". Through evaluation we were able to note the children's ease at adapting to this approach due to their familiarity with it, which had a beneficial effect on their understanding.

In terms of learning prowess, it is possible to conclude that scaffolding (Hammond, 2000) is of vital importance for developing children's knowledge. The results gained from evaluating the solution revealed that the advancements seen in levels 1 and 2 were more significant than that of level 3, as these levels more closely followed Hammond's premise of "supporting, consolidating and extending knowledge". Level 1 provided a grounding in basic operations on the number sequence and level 2 expanded upon this, allowing children to practice applying their knowledge in a familiar context. This had notable benefits in their ability to recognise how each question related to their existing knowledge and this led to an increase in the number of questions that were answered correctly. As we discussed in the evaluation, this improvement was also a result of the children 'getting used to' the software and thus the questioning style, as they were able to establish strategies for answering the questions. This was particularly apparent with level 3. In this level the same scaffolding as that offered between levels 1 and 2 was not given and this led to the conclusion that the 'jump' in difficulty and the difference in question style was too significant to continue extending the children's knowledge in a smooth and obvious pattern, where results were observed to reinforce this.

In terms of usability, an interesting result gained from the evaluation was the intuitive nature of the main menu. It was observed that having a 'central point' in the interface which the children could return to in order to start a new level or end their learning session formed a highly usable navigational structure, and was very easy for the children to pick up. It emerged that this was largely due to the familiarity and predictability related to this sequence of navigational actions. Accordingly we can offer the premise that having a central point in an educational tutoring system is a necessity for aiding usability.

By reflecting upon the approach taken to this project it is possible to recognise that users were the focus of much of the research, design and evaluation activities, which allowed us to more accurately understand their goals and expectations as well as their thoughts on the prototype solution and their reaction to its performance. This approach supported Carroll's task-artefact cycle (1991) as well as other HCI research concerning the importance of user-centred design. In particular the foundation stage teachers provided extensive feedback regarding the system's design as they led the discussion over the requirements validation and the redesign. This was appropriate as collectively the teachers offered a wealth of experience and thus were able to represent the views of the children. However, as stated during the design chapter it was not possible to gauge the views of the children as they were not able to comprehend how the low-fidelity paper-based representations related to computer based software, and accordingly they struggled to offer any particular justification to reinforce their views on the design. This may have been a result of presenting static information, which we identified as problematic for foundation stage children in the literature review because it is not easily interpretable. By not being able to involved children the feedback gained was somewhat limited, which may have contributed to the emergence of usability problems during the evaluation. Therefore a methodological advance for the future might be to note that providing a series of more usable computer-based prototypes is more appropriate for designing with young children. By using rapid prototyping to deliver limited functionality prototypes it might be possible to demonstrate particular aspects of the design in each iteration of development, from which we could assess the children's reactions and emerging needs and redesign as required. For the purposes of the project however, the advice offered by the teachers (which was grounded in their knowledge and experience of children) was sufficient for producing a solution that encompassed the knowledge gained through research in an appropriate and engaging manner.

A more general point about how to evaluate systems that can also be seen as a methodological advance regards how to evaluate tutoring systems. This is a contentious issue in HCI as there are not any specific guidelines that can be applied for this purpose. However, when evaluating the solution two methods were employed that achieved valuable results, which might be appropriate to apply when evaluating all tutoring systems. These were:

- The performance of children who had used the system, against those that had not. This examined the effect of the tutor for aiding the educational advancement of children in different conditions.
- The performance of the children at the start of the week, against their performance at the end of the week (before/after comparison, where children were tested before the evaluation period to form a baseline for control).

These methods provided a detailed set of results from which it was possible to assess the learning value offered by the system and its usability. However in order to draw more substantiated conclusions in relation to longer term usage we would need to test the tutor over a longer period of time and also evaluate its usability in more depth. This would allow us to recognise whether the tutor has the ability to sustain the benefits of accelerating development and also whether it is truly intuitive on first use. Moreover we would be able to assess whether the features that render the system as 'intuitive' lose their appeal over a sustained period of usage.

In preceding discussions, considerations that are important for developing tutoring systems that are usable, motivating and educationally valuable for children have been cited. In relation to this it is possible to note that the research area of children and computers is not particularly new in HCI, although the development of recommendations for designing 'child-friendly' systems is relatively under-developed and untested. Within this domain, Hanna et al (1997) offer guidance for assessing the usability of child-oriented software systems and Read (2004) has performed extensive research and analysis into how to design child-oriented software. With reference to these sources, the most important principles for designing software for children are:

- Strive for consistency
- Enable frequent users to use shortcuts
- Offer informative feedback
- Design dialogues to yield closure
- Offer error prevention and simple error handling
- Permit easy reversal of actions
- Support internal locus of control
- Reduce short term memory load
- Provide age appropriate instructions that are easy to understand, easy to remember, and supportive rather than distracting

These guidelines are notably quite generalised. Other than the last one, they all have an HCI focus and could quite feasibly be applied to any age group. However, through all research and evaluation activities it has been possible to recognise and reaffirm the importance of these principles, as they have a clear relation to salient issues regarding *general* usability that have already been discussed. In the context of child-related tutoring system it is possible to add additional detail though, and thus improve upon the level of existing theory for *designing educational systems for children*. Accordingly we can add three principles to the set offered by Read and Hanna et al:

- Computer based activities should be entertaining and relevant for children so that they are able to apply their knowledge in familiar, concrete scenarios.

- The principle of scaffolding should be supported so that tasks are provided that are supportive of the need to expand the scope of children's mathematics application.
- An appropriate and easily interpretable reward scheme should be provided to provide a local goal that children are motivated to achieve.
- The interface should a 'central place' for launching tasks so that it has a familiar starting point for initiating learning tasks.

These were noted as four essential features of the tutoring system that were key to the successes that were observed in the evaluation. They all have a firm grounding in both theoretical and practical research and have formed the basis of system design from both a mathematical and visual perspective with a view to making the experience of interacting with the system interesting and enjoyable.

In this project it has been established that computer-based tutors could form a natural part of the numeracy hour as they are capable of providing activities that challenge children and benefit their education both in a mathematics and IT sense. The future should encompass the development of the concept of computer-based tuition as a paradigm for learning so that they can be relied upon more heavily by teachers. However detailed consideration should be given to whether teachers actually want this, and if so, how much control they would like to exert over the performance of such a tutoring tool. We can conclude that computer-based tutors fulfil a role as an application for practicing applying classroom taught skills in ways that enable children to make associations with abstract concepts in concrete scenarios, so that they can become familiar with them. In accordance with the concept of scaffolding, these scenarios for practice should also expand the child's mathematical horizons. However, research has reaffirmed the importance of teachers as mediators for educational development and as such their position cannot be considered as being under threat.

7.1 The Future

It is important to consider *the future* in any project or design activity as new research and new methods may help a user to perform their tasks at a higher level or with greater effect and ease. Moreover through further consideration of the problem domain we may be able to recognise ways in which the solution can be developed to improve upon the user's interactive experience or support more of their requirements and goals.

A clear area for future work would be to implement the design solutions that were proposed as methods to solve the usability issues that became apparent in the evaluation chapter. This would allow the system to be refined in light of the results gathered so that it is capable of supporting the preferences shown by users for performing their tasks. This complies with the development cycle that was discussed with the software designers in the requirements analysis chapter, such that after evaluation it was common for a product to undergo a further phase of redesign before release.

Another direction for future extension to the project is to integrate more lessons into the system. These could cover different mathematical concepts or even different subjects of the foundation stage curriculum to extend the scope for the child to practice applying their skills. Moreover these new lessons could use different learning metaphors for different types of lesson in order to maintain the level of interest and motivation that was prevalent in the evaluation of 'frog in the pond'. In relation to this, the house of games metaphor could be developed to encompass other tasks. Through extension of this metaphor the 'frog in the pond' task could be launched from a button in the garden, whilst another game could be 'the spider in the attic' or the 'dog in the garden' (for example) which could also be launched from appropriate areas of the house. This would clearly draw further parallels with real-world contexts that children have experience of and find readily interpretable.

An interesting development of the solution would be to allow the teacher more freedom to customise the task. By providing the teacher with the freedom to input audible instructions or feedback, or indeed remove it, they could tailor the tutor to suit particular lessons of the classroom and thus more accurately reflect the style of instructions or feedback being used in the classroom. This might have the beneficial effect of reducing learning time picking up how to use the system, due to its familiarity. It might also enable children to more rapidly pick up how to use mathematical language in appropriate contexts.

The teacher could also be allowed to alter actual aspects of the task to test particular parts of the child's understanding of mathematics and also literacy. This may have the effect of enabling the teacher to rely more heavily on the tutor to truly support personalised tuition of mathematical concepts, whereby they are able to control the particular skill that a child needs to work upon in a learning session. An example might be to change the identity of the question so that the user must answer questions of the form; "1 more than x is 5, what is x?" Or to remove the interface persona, and thus force the user to interpret text instructions. However, with reference to the results observed in the Evaluation, it was noted that the children enjoyed working with the persona and relied on its instructions to interpret the question. Accordingly we can recognise that it would be necessary to research this area for future work in more depth, so that the learning benefit of the tutor was not detracted by removing salient parts of it.

The emergence of new technology is an exciting area for future work on the project. In the context of this project the emphasis should certainly be on supporting the activities of the classroom and thus enabling children to access the functionality of the tutorial without having to overcome usability problems that exist in the technology behind it. This relates to the current developments in ubiquitous computing, which propose that technology can eventually become so transparent that "there will be a shift from thinking about computers as extensions of the desktop to extensions of the environment that we live in" (Abowd et al, 1999). Where children are concerned, this is very important given their computing experience and control over input devices, which have formed the constraints for many of the design decisions throughout this project.

Touch screen systems that allow a child to physically act upon objects in the interface and enter their answers would clearly reduce the hindrance of current computing hardware, particularly input devices. This could make the child-computer interaction more transparent and intuitive and also allow for more far ranging learning metaphors to be used. It would also support the usability requirement 21, which stated the importance of allowing a child to manipulate objects in the interface in order to arrive at their answer. By physically moving objects around the screen clear parallels can be drawn with activities of the classroom which may help to reduce the complication of children mapping their thoughts to computer input. As we discussed in the literature review, Norman (1998), described the phenomenon underpinning this mapping as "cognitive engineering", and he stated that where the gap in interactive and communicative needs between humans and computers can be reduced, the system is inherently more usable and capable of supporting a user's goals; thus benefiting the child's experience of interacting with the system.
In relation to this, large touch screen displays could also help to support collaborative work, as any number of children could use the system and all could partake when calculating answers and manipulating on-screen objects. The DFEE (1999) state the importance of group work as a method for learning as research has shown that discussion and collaboration can help to reform educational misconceptions and also build personal relationships in the classroom. Moreover, given the limited computer hardware available to infant schools, this may be a more efficient method for all the children to gain experience of ICT media in a way that is educationally valuable, but also enjoyable.

Voice recognition is another emerging technology that has huge potential in reducing the disparency between humans and computer input as allowing a child to simply *talk* to the system when entering their answer could have beneficial results in terms of the child's understanding of the system. It might also provide opportunities for them to practice using mathematical language in appropriate contexts. However, we can recognise that this may detract from the advantages offered by the current system where text, visual and audible information provide a breadth of means by which to input and receive output that are all important mediums for communication that children need to gain experience of. Also problems could emerge where input and output are confused or where the reliance on speech could disrupt other children in the classroom. Moreover, current voice systems require extensive user training which of course would not be appropriate given the need for the system to be highly learnable. Clearly then, in its current form voice recognition systems offer no clear benefit over current tutoring solutions, but in light of future advancements they may emerge as a viable alternative.

In this analysis of areas for future work we have noted a number of interesting extensions for the current prototype solution, some areas certainly being more relevant than others. The most exciting of these in my opinion would be to develop the current solution so that it is more customisable for the teacher. We have identified the potential for teachers to be able to tailor the instructions, feedback or indeed actual aspects of the lesson itself in accordance with their lessons, and I think that this extension is most relevant given the need for children to practice applying their existing knowledge in ways that they understand and can easily relate to.

8 Bibliography

8.1 Books

- Ager, R., (1998) *Information and Communication Technology in Primary Schools*. Fulton
- Bruner, J., (1967) *Studies in Cognitive Growth*. Wiley.
- Cooper, H., Sixsmith, C., Foster, M., Foster, R., Saunders, P., Elliot, L., (2003) *Teaching across the Early Years 3-7: Curriculum Coherence and Continuity*, edited by Hilary Cooper and Chris Sixsmith. RoutledgeFalmer
- Copeland, R.W., (1979) *How Children Learn Mathematics: Teaching Implications of Piaget's Research*
- Dix, A., Finlay, J., Abowd, G., Beale, R., (2000) *Human Computer Interaction*. Pretice Hall
- Donaldson, M., (1978) *Children's Minds*. Fontana.
- Eysenck, M. W., Keane, M. T., (1995) *Cognitive Psychology : A Student's Handbook*. Hove, Erlbaum.
- 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.
- Hughes, M., (1986) *Children and Number: Difficulties in Learning Mathematics*. Basil Blackwell, New York.
- Keller, J.M., (1983) *Motivational Design of Instruction: Instructional-design theories and models: An overview of their current status.* Hillside, NJ, Erlbaum.
- Landauer, P., Helander, M., (1997) *Handbook of Human-Computer Interaction.* Elsevier Science B.V
- Liebeck, P., (1984) *How Children Learn Mathematics: A Guide for Parents and Teachers*. Penguin Books
- Moore, A.,Redmond-Pyle, D., (1995) *Graphical User Interface Design and Evaluation*. Prentice Hall
- Mullet, Sano (1995) *Designing visual interfaces*.
- Newman, W.M.C., Lamming, M.G., (1995) *Interactive System Design*. Addison Wesley.
- Norman, D.A., (1998) *Design of Everyday Things*. MJT Press
- 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.
- Riding, R., dan Rayner, S., (1998) *Cognitive Styles and strategies*. David Fulton, London.
- Shneiderman, B., (1998) *Designing the user interface: strategies for effective human-computer interaction.* Addison-Wesley.
- Sommerville, I., (2001) *Software Engineering 6th Edition*. Addison Wesley

8.2 Websites

- Abowd, G.D., Atkeson, C., Essa, I., MacIntyre, B., Mynatt, E., Potts, C., Ramachandra, K., Ribarsky, W., Rugabar, S., Starner, T., (1999) *Augmenting the Capture of Understanding of Everyday Experiences* [WWW] http://citeseer.nj.nec.com/18221.html (May 5, 2004)
- DfES/Becta (2002), ImpaCT2, *The impact of information and communications technologies on pupil learning and attainment*. Schools research and evaluation series. [WWW] http://www.becta.org.uk/page_documents/research/ImpaCT2_strand1_bw.pdf (December 07, 2003)
- W3 Schools (1999), *Introduction to Flash*. [WWW] http://www.w3schools.com/flash/flash_intro.asp (February 29, 2004)
- Hitcham, R., (2000) *The use of ICT in the Mathematical Development area of the Early Learning Goals*. [WWW] http://www.hitchams.suffolk.sch.uk/foundation/numeracy/index.htm (November 22, 2003)
- Macromedia, (2004) *Macromedia Flash MX 2004* [WWW] http://www.macromedia.com/software/flash/?promoid=home_prod_flash_082403 (April 11, 2004)
- Nielsen, J., (2001) *Heuristics for User Interface Design.* [WWW] http://www.useit.com/papers/heuristic/heuristic_list.html (February 22, 2004)
- Read, J (2004) *Human-Computer Interaction and Children* [WWW] http://www.chici.org (May 5th, 2004)
- Sun Microsystems (2004) *Java 2 Platform, Standard Edition* [WWW] http://java.sun.com/j2se/ (April 11, 2004)

8.3 Articles

- Hanna, L., Risden, K., Alexander, J., (1997). *Guidelines for usability testing with children*: Interactions May
- Rudd, J., Stern, K., Isensee, S., (1996) *Low versus High-Fidelity Prototyping Debate*: Interactions, January

8.4 Journals

- Anglin, G.J., (1995) *Instructional Technology*. Anglin, Englewood, USA
- Bryant, P.E., Trabasso, T., (1971) *Transitive inferences and memory in young children*, Nature, vol 232, pp. 456-8
- Carroll, J.R., Rosson, M.B., (1991) *Deliberated evolution: Stalking the view matcher in design space.* Human-Computer Interaction, vol 6, issues 3 and 4, pp. 281-318
- Hinostroza, J.E., Mellar, H., (2002) *Pedagogy Embedded in Educational Software Design: Report of a Case Study*. Computers and Education, vol 37, issue 1.
- Markopoulos, P, (2003) *Interaction Design and Children*. Interacting with Computers, vol 15, pp. 141-149
- Newton, D.P., (2001) *Helping Children to Understand*. Evaluation and Research in Education, vol 15, issue 3, pp. 119-127

- Nielsen, J., (1990) *Paper versus Computer Implementations as mockup scenarios for heuristic evaluation.* International conference on Human-Computer Interaction, vol 3, pp. 1-8
- Pappas, S., Ginsburg, H.P., Jiang, M.Y., (2003) *SES differences in young children's meta-cognition in the context of mathematical problem solving*. Cognitive development, vol 18, issue 3, pp. 431-450
- Rich, E., (1979) *User Modelling via Stereotypes*. Cognitive Science, vol 3, pp. 329-354
- Rohrbeck, C.A., Ginsburg-Block, M.D., Fantuzzo, J.W., Miller T.R., (2003) *Peer-assisted learning interventions with elementary school students: A meta-analytic review*. Journal of educational psychology, vol 95, issue 2, pp. 240-257
- Segers, E., Verhoeven, L., (2002) *Multimedia Support of Early Literacy Learning*. Computers and Education Journal, vol 39, issue 3, pp. 207-221
- Zin, N.A.M., Zaman, H.B., Noah, S.A.M., (2002) *Multimedia Mathematics Tutor: Matching instruction styles to student's learning styles*. International conference on computers in education proceedings, vols 1 and 2, pp. 1433-1434

8.5 National Curriculum References

- DFEE, Department for Education and Employment, (1999):
    - The National Numeracy Strategy: Framework for teaching mathematics.
    - Curriculum guidance for the foundation stage

APPENDICIES

- Literature Review Appendix
- Requirements Appendix
- Design Appendix
- Implementation Appendix
- Evaluation Appendix

Literature Review Appendix

1. Design Heuristics

1 Design Heuristics

The most important design heuristics that are relevant to the tutoring system are given below, with reference to Dix et al (2000).

Learnability - those features of the interface that allow intuitive use. Learnability dictates that the interface provides features that allow novice users to understand how to use it and then how to develop their understanding further.	**Predictability**	Based on their past interactions with the system, or by having performing similar actions, the user is able to determine the effect of a particular action.
	Synthesisability	The extent to which a user can apply their knowledge of previous interactions in order to predict the behaviour of the current state of the system.
	Familiarity	The extent to which a user can apply their knowledge of other interfaces or domains within the system.
	Generalisability	Support for the user to extend knowledge of specific interaction within and across applications to similar situations.
	Consistency	The input-output behaviour arising from different situations should be the same throughout the system, so that the user is not "surprised" by the results of their actions. This relates to use of colour, layout, fonts, and so on.

Flexibility – refers to the multiplicity of ways that the user can exchange information with the system	**Substitutivity**	Allowing equivalent values to be substituted for each other (over input and output) so to avoid unnecessary user calculations. This helps to minimise user error and cognitive load.
	Customisability	The extent to which the interface is modifiable, to suit the user.

Robustness – those features of the interface that support the successful achievement and assessment of goals.	**Observability**	Allowing the user to evaluate the internal state of the system based on the information presented to them at the user interface.
	Responsiveness	How the user perceives the rate of communication with the system
	Task Conformance	The degree to which the system supports all of the tasks a user wishes to perform in the domain, and if it supports them in the desired way.

Requirements Appendix

1. Evaluation of Existing Tutoring Systems
2. Requirements Document

1 Evaluation of Existing Tutoring Systems

1.1 Teddy Bear's Picnic

Teddy Bear's Picnic is developed by Sherston Software group, for children aged 4-7 years. It provides a standalone tutoring environment with 7 activities that allow children to apply their literacy and numeracy skills, whilst simultaneously developing their IT skills. The system is developed using Macromedia Director and thus offers a high level of graphics in highly interactive environment. The system is run using an executable file.

The system launches to fill the screen, meaning that the user cannot click on any buttons outside the scope of the interface. This is clearly advantageous with child users, as they can only perform actions in the task domain. A small range of colours are used throughout the interface, and simple graphics and concise use of text convey instructions and feedback, thus complying with Nielsen's (2001) usability principle stating that interfaces should be aesthetic and minimalist in design.

On entry to the system the user can select their name from a list (pre-registered by the teacher). From observing children in the classroom it was possible to recognise that they can all recognise and pick-out their names from a list thus making this feature appropriate given their literacy skills. When the children select their name from the list, a user profile is initialised that records the activities the child has attempted and the results they have achieved and allows the system to personalise all instructions and feedback. All user profiles can subsequently be accessed by the teacher through the 'teacher control area', where a summary of the activities attempted and the percentage of questions answered correctly are detailed. The teacher is able to print the results out if required. This feature of the system relates to the need for the system to provide a printable summary, as specified by the foundation teachers. However we can also recognise that this feature reduces the system's ability to support more than one child when using the system, due to the personalised feedback that it gives. The need to support multiple users was also cited as an important feature for the system, and thus we can recognise that using profiles may not be appropriate.

The system has a main menu that displays the 7 tutorial activities (refer to figure 1 at the end of this section). The menu is clearly laid out using iconic buttons, where text and pictures are used to form a large 'hotspot' for each activity, which is very easy to 'hit'. The iconic buttons are spread across the interface making full use of the space available. The menu has a teddy bear interface persona placed in the bottom right-corner. The persona welcomes the user and provides the instruction that they should choose an activity. When an activity is chosen the persona indicates the choice made by providing audible feedback of the user's action.
The menu has an orange background which is consistently used throughout the interface.

From the main menu there are a set of options that can be configured by the teacher to customise parts of the system as well as the appearance of the interface. This area is password protected, meaning that only the teacher can access it. The teacher is able to specify:

- The volume of the sound output from the tutor.
- The difficulty of the tutorial content: level 1, 2 or 3. The level has the effect of increasing the difficulty of the task, as well as changing the required IT skills to complete the task. Increasing the difficulty of the task itself can be seen as advantageous as it allows the tutor to support the children's increased level of understanding. However, changing the mode of interaction, from point-and-click, to

drag-and-drop for example (refer to figure 2) is inconsistent, and is a potential area for usability difficulties, given that the related set of instructions do not detail the changes.

- The number of times a user should repeat an activity before it ends. In the paying for shopping game the user must complete 3 sums per round. Thus the teacher can specify whether they repeat a round 1, 2, or 3 times.
- The teacher can create a user profile for each user. As previously explained, a profile allows the user to log-in to the system so that the activities they attempt and their relative success are recorded.

The paying for shopping activity is the most relevant in terms of mathematics tutoring (refer to figure 3). The task involves the child clicking on the shopping bag to empty its contents, and subsequently paying for each item. The user pays for each item by clicking on a coin from a choice of four. Instructions for the activity are provided solely by the interface persona, without any use of text, graphics or animation. The user can click an exit button in the corner of the interface to quit the task at any time and return to the main menu.

Advantages

- The instructions given by the interface persona are very clear and concise. The persona describes how the user should interact with the interface and what the goal of the task is. The pace of delivery is quite slow so to allow users to listen-to and pick-up vocabulary, as used in correct contexts.
- Rapid feedback is given over by the interface persona to denote whether the child has answered the question correctly, denoting a high level of system responsiveness (a desirable design principle stated by Dix et al, 2000). If the answer given is correct, a brief animation demonstrates the money entering the till and the item going into the shopping bag which provides some satisfaction for the user.
- The system has a consistent look and feel which aids its familiarity and thus promotes learning and generalisation throughout the interface.

Disadvantages

- The instructions given by the interface persona make the pace of the task very slow. They are repeated for every question of the task during which time the user is ‘locked out’ of the interface (they cannot interrupt the instruction to continue the task). This denotes a low level of synthesisability (the user cannot “learn from doing” given a set of tasks) and also has implications for removing user control over the task.
- There is no option to repeat the instructions on demand. Although the instructions are repeated for every question, the user cannot repeat the instruction on demand. This is disadvantageous as there is no text to reinforce the instructions and thus there is no permanent media in the interface for communication with the user.
- If a user answers a question wrongly, the correct answer is simply revealed without any opportunity to have a ‘second chance’ to reform the mistake. This has implications over the learning value of the task, as the user can seemingly never answer a question wrongly.
- At the end of the task the set of questions is repeated without asking the user if they actually want to repeat them. As we have discussed the number of times a round of questions is repeated is an option specified by the teacher in their set of controls, but it clearly has implications over the user’s level of control over the course of the task.
- When a set of questions are repeated they are repeated at the same educational level, providing no means for the user to explore more difficult applications for their knowledge. This relates to the design principle of task conformance such that the system does not effectively support self-directed learning for users, and thus may not enable them to achieve all the goals they desire. As we have seen, the teacher is only able to increment the level of difficulty of the task through the set of teacher controls; a user cannot do this independently.

- Increasing the difficulty level also impacts the direct manipulation skills a user must apply to interact at the interface. At the first level the user must point and click to select a coin and place it in the till, in subsequent levels 'sticky mouse' and drag-and-drop operations are introduced, without any warning or instruction. This has implications over the predictability and consistency of the system (refer to screen shot at the end of this section).
- There is no reward scheme or aim for completing the task. The user's success in each task is logged in the user's profile, but this is invisible, and thus it is difficult to grasp any level of satisfaction from completing the task.
- There is no indication of the user's progression through the task which means the user cannot gauge how many questions are left in the task set. Dix et al describe this as observability, which is clearly demonstrated at a low level as the information presented at the interface does not provide enough clues on the internal state of the system.
- The 'exit' button, used to quit a task and return to the main menu is labelled with an arrow, which the user would expect to return them to the previous question. Thus there is the potential for the user to misinterpret the function of this button, and subsequently be surprised by the action performed when it is clicked.

1.1.1 Associated screen shots

Figure 1: The main menu for the system. The persona is the teddy bear in the bottom-right corner of the interface.

Figure 2: The teacher control area. The change in difficulty associated with each level is described in each case.

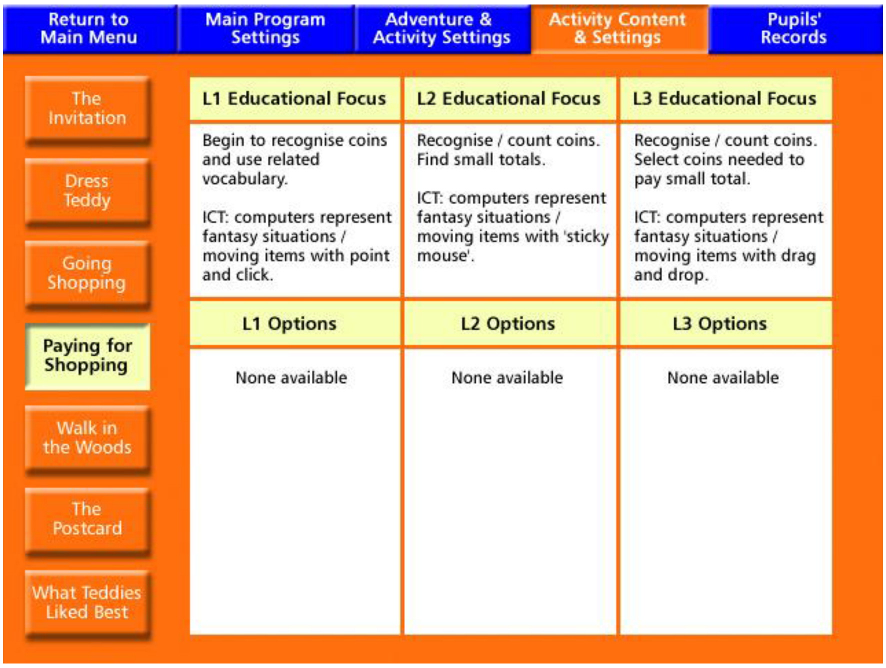


Figure 3: The shopping activity. The 'back' arrow allows the child to return to the menu of activities.

1.2 Tizzy's Toybox

Tizzy's Toybox is also developed by Sherston Software group for children aged 4-7. Similarly to Teddy Bear's Picnic (also developed by Sherston) the system is developed using Macromedia Director to enable a highly visual interactive environment. The system is standalone (is run using an executable) and provides 10 activities for children to practice applying their numeracy, literacy and ICT skills.

The system launches to fill the screen, meaning that the user cannot click on any buttons outside the scope of the interface. However, the size of the task domain within this area is very small at approximately 400x400 pixels which makes poor use of the available space and has a negative impact on the size of the task related objects and text. As we have seen in requirements analysis chapter, the size of objects and text are particularly important factors in affecting a child's ability to interpret a task with any success.

The system has a main menu (refer to figure 4 at the end of this section) which displays all 10 activities available to the user. The menu is laid out symmetrically with 5 activities displayed in two columns, making good use of the available space. Activities are launched via text hyperlinks, although there is an associated picture for each activity. The picture seemingly forms part of the 'hotspot' for launching the activity but it is actually redundant which is not clearly depicted through the interface. Consequently the user only has a small hotspot for launching each activity.
The menu has a clown interface persona, who instructs the user to choose an activity from the menu. This message is reiterated with very concise text output, which remains on the screen and thus gives some permanence to the instruction. When an activity is chosen it is underlined and the persona reaffirms via speech output that it has been selected providing verification of the user's action.

The user can access a set of 'help' instructions from the main by clicking on the interface persona (refer to figure 5). These instructions are delivered using text and detail three basic actions that the user must understand in order to begin using the system. From previous discussions we can determine that text might be inaccessible for child users because of their limited ability to decode it. Also all other parts of the interface use the interface persona to deliver instructions and feedback via speech output, using text to reinforce key points. Thus we can recognise a point of clear inconsistency and potential confusion for the user. Moreover the button that returns the user to the main menu is simply labelled 'OK'. Although you would usually expect to return to the previous point of the interface after reading help instructions (Dix et al, 2000), a label of this nature is highly unintuitive as it gives no clues about the navigational action performed by clicking on it.
Help systems are a contentious issue in HCI as it is argued that help should not be needed if a system is designed to accurately support its user's tasks. With child users this issue is especially salient, as we must consider that the child may not know what 'help' means, or they might not be able to map the psychological thought process that they need help with accessing the relevant information provided by the system. Furthermore, if they do manage to make this mapping we can recognise that they might not understand the instructions provided, especially when they are provided in text format.

From the main menu there is a further link to set of teacher controls, allowing the teacher to customise the level of task difficulty, the text highlight colours and the volume of sound output (refer to figure 6). The teacher can also exit the system from these set of controls, which is an option not available in any other part of the interface. This is advantageous as it provides some protection from a child user exiting the system mid-session.

The link to the teacher controls is not protected from children accessing it however, as it was in Teddy Bear's Picnic.

The counting activity is the most relevant in terms of mathematics tutoring (refer to figure 7). The task involves the child clicking on each mouse, which is then moved to be alongside one of the cats. The interface persona counts each mouse the user clicks on allowing them to listen to the number sequence. The task is performed 3 times, with a varying number of mice to click on. The user can click an exit button in the corner of the interface to quit the task at any time and return to the main menu.

Advantages

- The interface persona gives all instructions and feedback throughout the interface, using speech output. Text is used to reinforce salient points giving some permanence to the instruction.
  The user is able to repeat instructions or feedback on demand, by clicking a 'lips' button. This is a good feature as it gives the user some control over the delivery of the instructions.
- The user is 'locked-out' of the interface for the initial instruction set given by the interface persona meaning they have to listen to it. However subsequent instructions and feedback can be interrupted by the user allowing them to exert a high level of control over the learning session.
- The user can quit the task at any time and return to the main menu using a Toybox iconic button, thus relating to Nielsen's usability principle of user control and freedom.
- The user can only exit the system in the teacher options area, which reduces the chance of a child accidentally closing the program (refer to figure 6).
- Rapid feedback is given during the task, demonstrating a good level of system responsiveness. The interface persona has a high level of participation in the task (for example it counts through the numbers with the child) which makes the system very interactive.
- The system has a reward scheme in the form of an animation which is performed at end of the task. The persona performs a different circus routine every time the user completes a task, providing some interest. However, we can recognise that the animation has little to do with the relative achievement of the user.
- There is a consistent look and feel to the interface. The colour scheme is very simple and attractive and all buttons are consistently placed in the interface throughout the task, aiding its familiarity.
- The system's menu structure is good. The consistent placement of the buttons to navigate the interface and the consistency of the action performed by clicking on them mean that the user is never surprised by 'where they are' in the interface.
  However, a notable poor aspect of the task was that a user cannot to return to previous sum.

Disadvantages

- The interface persona operates *in* the task domain but provides no *real* assistance to the user in the task. Thus it is not clear that the user cannot interact with the persona which can be confusing.
- There is a low level of system observability (Dix et al, 2000) and thus the user is unable to grasp how far they have progressed through the task.
- The lips button (used to repeat the interface persona's speech output) and the Toybox button (used to quit a task and return to the main menu) are poor in terms of their usability. Although the functionality of these buttons is explained in the help menu, their icons do not accurately represent the action performed by clicking them. This means that the user must *recall* the button's functionality instead of being able to simply recognise it. Clearly this adds to the user's required memory load.

- The teacher must increment the system's level of difficulty. The means that there is little scope for a user to expand their knowledge in a self-directed learning session and thus challenge their understanding of a given concept.
- There is no summary of questions answered when a user completes a task, meaning the user cannot print a copy of the questions they have answered. As specified in the requirements analysis chapter, there a need for some hard-copy feedback to be given by the tutoring system, for the needs of the foundation stage profile, and thus the system clearly does not support the user's needs.
- The tasks were very short at 2-3 questions. This provides little scope for the user to practice applying their knowledge for any prolonged period of time before having to try another activity.

1.2.1 Associated screen shots

Figure 4: The main menu. The clown is the system's interface persona.

Figure 5: The help menu. A key point to note here is that the clown with the 'next' button cannot be clicked on, whereas the clown with the 'ok' button can be.

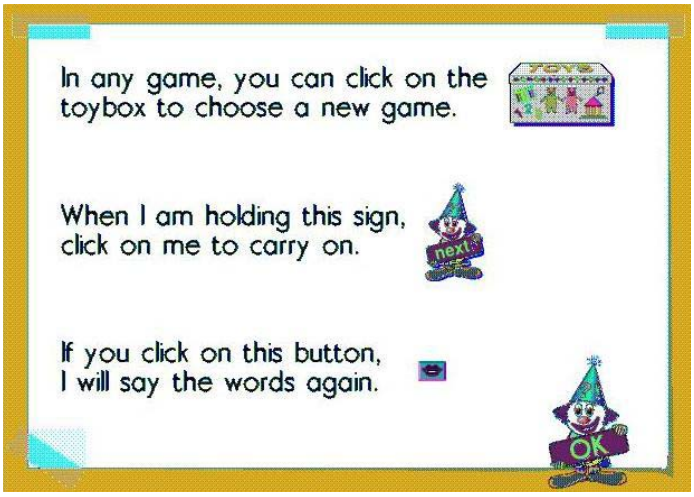


Figure 6: The teacher control area, showing the customisable aspects of the interface, and the only point of exit from the system.

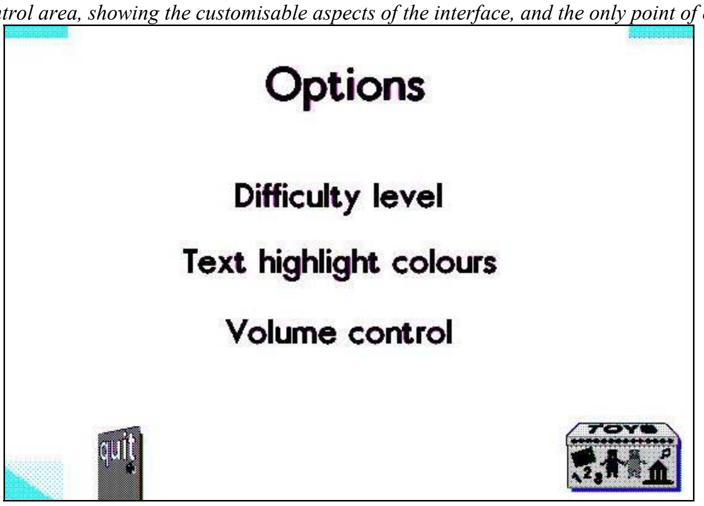
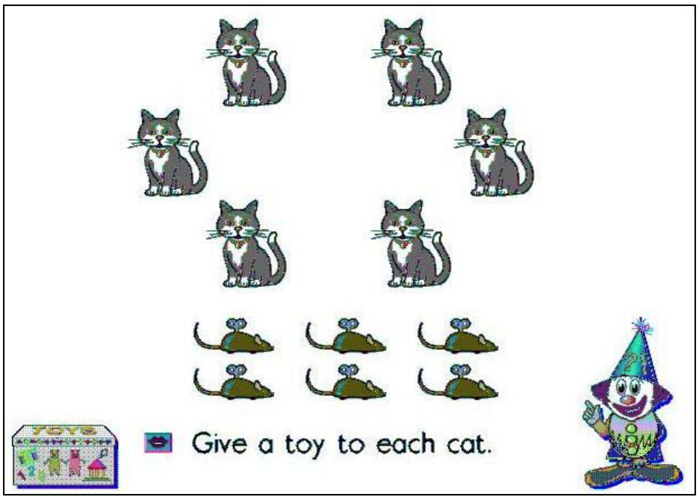


Figure 7: The counting activity, note the interface persona's direction of the task.



1.3 BBCi website

The BBCi website offers a set of 6 numeracy activities for children aged 4-6 years. The activities are delivered as Macromedia Flash movies, offering a highly visual and interactive task environment. Being a website the tutorial requires a web browser to run, in this case Internet Explorer v6.0.

The website uses a main menu to display the 6 numeracy activities (refer to figure 8 at the end of this section). The menu makes poor use of the space in the screen by displayed tasks as text hyperlinks at the left of the screen. The hyperlinks have bullet points that are totally inactive as links to the activities that they are associated, and thus the user has to click the text part of the hyperlink to navigate the interface, meaning the 'hotspot' is very small. The hyperlink changes colour to signify that the link has been visited. This is not explained in the interface and would potentially not be understood or appreciated by child users. Moreover the colour of the hyperlinks against the green interface background is not a good colour pairing in terms of readability.
A 'cat' is used as a basic interface persona for the menu to state (via a textual speech bubble) that the child should choose an activity. The cat does not give any speech output and the user cannot interact with it, and thus we can recognise that it is largely irrelevant to the interface.
The menu has a green background, which is used consistent throughout the interface to denote the user is in the BBCi 'tutorial area'. This is especially important with a website like BBCi as there are many interface screens covering a range of topics.

The tutorial area of the website is enclosed with the BBCi web-frame. The frame has many links to take the scope away from the tutorial area and move to other web pages like the BBCi home-page, or the copyright terms page for example. This is a concern for child users as they could accidentally click one of these buttons and subsequently be 'lost' within the BBCi website, without the knowledge of how to use the web browser 'back' button to return to the tutorial area. Also there is a banner at the bottom of each webpage denoting that the user requires the 'shockwave player' to view the activity. This is distracting to the task, and further to the web-frame it is another hyperlink to take the scope away from the tutorial area, which is a potential problem for child users. Although Macromedia now claim that "96.4% of web users can experience Macromedia Flash content without having to download and install a player", it is important that the user should not have to download additional software to run the tutoring system, for usability reasons. Typically inexperienced internet users can be 'scared' of downloads and their implications over system safety. Requiring users to download a program also requires an assumption over their internet connection which cannot be made in all cases.

When the user chooses one of the 6 numeracy activities the 'work area' opens. This forms the task domain where all actions related to the task are performed. However, the size of the task domain is very small in relation to the size of the screen (approximately 400x300 pixels), which makes very poor use of the available space in the interface. This has an effect on the size of the text, buttons and graphics that are displayed within the task which could have implications over a child's ability to interpret parts of the task effectively.

Each activity within the system has two standard features:

- A set of teacher controls for controlling the task. Clicking this link opens a control menu where the teacher is able to specify:
  - The difficulty of the tutorial content: level 1, 2 or 3. The level has the effect of increasing the numbers used in the task. For example in a task over addition level 1 involves adding numbers in the range 1-5, level 2 also includes numbers 6-10 and level 3 also includes numbers 10-20. This is an advantageous system control as it allows the teacher to use the tutor throughout the foundation year to support the children's increased understanding of a mathematical concept. Also,

when the user completes a task, they can optionally move to the next level themselves, providing the scope for self-directed learning.
    - Whether a timer will be used in the activity. This adds a speed element to the task.
    - Whether text output is to be used. This allows the teacher to specify whether text is used as part of the instruction set. As we have discussed, foundation stage children would not benefit significantly from this control being switched on due to their limited ability to decode sentences of text.
- 'How to play' instruction set. There are instructions for each activity to guide the user through each part of the task. The instructions are delivered in the task domain where the interface persona and related graphics are used, to promote familiarity when the user begins the real task. However, the instructions are completely separate from the task itself which makes them difficult to remember. Also the interface 'locks the user out' from exiting or skipping parts of the instruction set, which has clear implications over removing user control.
  The instruction set is also very long and details all the knowledge required to complete the task. In an intuitive environment, the user should only need to know the basic essentials of how to get started in the task and from there progression should be obvious. If they require further guidance they should be able to access it.

An interesting numeracy activity is the Dartboard game, which is the most sophisticated of the 6 available (refer to figure 9). This task involves the user using a dart to hit a number on the dartboard in relation to instructions given over by two interface persona, one who speaks and one who assists the activity. The motivation to play the game is to accumulate gold coins. If the user reaches 10 coins (one gold coin is rewarded per right answer) they progress to the next level (which is more difficult).

Advantages

- The interface persona tells the user which number to hit on the dartboard, given a choice of 7 numbers. The persona uses speech to convey the number required, which is also displayed in the corner of the screen. This provides two clear mediums for communicating the salient information of the task to the user.
  The user is able to interrupt the persona's instructions when placing the dart, allowing them to skip the instructions that are common for each question in the task. Thus the user has a high level of control over the pace at which they perform the task set.
- The task incrementally increases in difficulty which is an effective way of expanding the child's knowledge base. Introducing the task with simple questions, requiring the application of basic mathematical knowledge in the first instance allows the user to establish an understanding of how to manipulate the darts within the task. Subsequent increases in difficulty help to maintain the challenge of the learning session.
- Feedback was given by the interface persona both whilst performing a sum, and after completion, clearly complying with Nielsen's usability principle over visibility of system status, and contributing to the system's high level of responsiveness. Feedback is given using terms like "too many" or "too few" which gives the child some quantitative clues about their mistake, providing the scope for rethinking the situation. The user is then given a second chance to get the answer right, before the task moves onto the next question.
- The reward scheme of accumulating 10 gold coins gives an incentive for completing the task. The user must accumulate a total of 10 coins before the task ends and they can move up to the next level. This provides a local goal, and denotes a good level of interface observability, such that the user can assess their progress towards their goal whilst performing the task.
- At the end of a task the user can chose to repeat the task at the same level, move to the next level, quit and return to the main menu, or print their score sheet (refer to figure

10). This enables the user to maintain control over their learning session, relating to Nielsen's usability principle of user control and freedom. The score sheet denotes the questions answered, the answers given and whether it was correct or incorrect in a simple table.

Disadvantages

- There are two interface personas. The main persona delivers instructions and feedback very successfully. However the second persona acts as an assistant who seemingly has no role. This is unrelated to the task and is a potential point of distraction for the user.
  Also the main persona had a tendency to 'giggle' which is very repetitive and potentially annoying.
- The main interface persona's instructions can optionally be reiterated using text (an option in the teacher control area). This text is too small and scrolls across the screen. This has implications over whether the child could decode the text at all, and also whether they could do this before the text disappeared off the screen. A better method would have been to use a concise amount of text (key words) that stayed on the screen, as seen in Tizzy's Toybox.
  A further disadvantage of the instruction delivery is that the user cannot repeat it on demand, which is disadvantageous if they misunderstood part of the instruction or simply missed part of it.
- The direct manipulation operation required to place the darts is not intuitive. The user has to use a click-and-click operation to first select the dart and then click on the dart board. However there is no verification that the dart has been selected and thus it is difficult to recognise whether the task is being performed correctly. The click-and-click mode of interaction is not explained in any of the instructions also, and thus it is difficult to grasp for novice users. A more intuitive mode of manipulation would be to drag-and-drop the darts on to the dartboard as the user could visualise the darts moving towards the dartboard, thus eliminating the aforementioned disadvantages.
- The number given as the aim for the task (the number the user must hit using the dart) is placed in the corner of the screen and flashes, which could cause distraction for the user.
- If the user gets a question wrong twice, the interfaces simply moves to the next question without revealing the correct answer. Revealing the right answer could potentially help the user to recognise their mistake.
- The incremental increases in difficulty involved an unusually big step up in difficulty. Level 2 requires the user to hit 'doubles' on the dart board, which requires knowledge of multiplication. This is not a lesson of foundation stage mathematics.
- The exit button to leave the task is very small and is labelled 'back'. This has implications over whether it is understood to return to the previous sum, or to the main menu. Moreover when it is clicked the scope returns to the main menu of the system, not the menu for the dartboard game. There is no other means by which to return to the game menu, which signifies a poor system menu structure and is is frustrating for the user.

1.3.1 Associated screen shots

Figure 8: The main menu for numeracy tutorial activities. Note that the cat persona here provides now audible instruction to the user

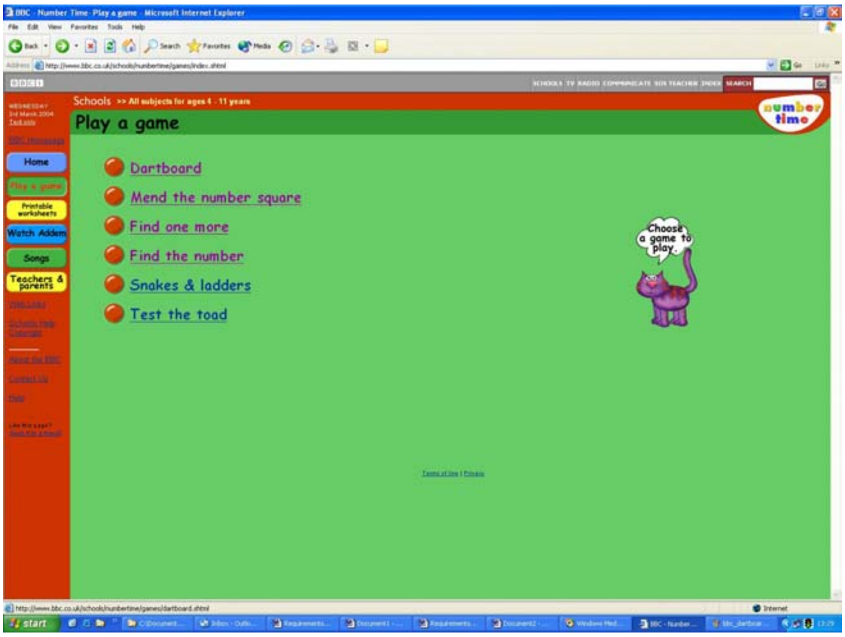


Figure 9: The dartboard game. Note the small task domain, given the available space in the interface.

Figure 10: The post-task summary sheet, allowing a view a summary of the task they have completed.

2 Requirements Document

2.1 Introduction

This section draws together the key requirements gathered from the literature review (section 2), the stakeholder-centred requirements analysis (section 3) and the existing tutor evaluations (section 3). The knowledge gained from these sources provides a greater understanding of the potential design space. By focussing this knowledge through appropriate requirements we can identify the key tasks and objectives for the tutoring system and thus more effectively support the ultimate goal of aiding children's development and understanding of mathematical concepts.

The requirements detailed in this section comply with Preece et al's (2002) framework for structuring a requirements document. It is important to note that these requirements form the scope for a complete mathematics tutoring system and have been prioritised in order of importance.

Modifications to the requirements that were suggested in the requirements validation process with Louise Aylott are marked 'user related modification' and are explained where relevant.

2.2 High-level Requirements

Further to the framework provided by Preece et al (2002), it is necessary to state a number of high level requirements that the system should meet in order to support the user's tasks and needs:

- To support learning and teaching processes: The system should be an effective mathematics teaching support tool for children at the foundation stage.
- To provide a challenge: The system should allow child users to apply classroom taught skills in suitable contexts. Tasks should be of an appropriate level of difficulty such that they support the user's mathematics development.
- To provide a highly usable and intuitive user interface: The user interface should be tailored to the needs and abilities of 5 year old children, in consideration of their computer literacy.
- The system should be capable of running on the school computers.

2.3 Functional requirements

These requirements capture the functionality that the system should provide:

1. The system should display a complete set of the available task options and customisability options for the user.
   <u>Origin:</u> Literature Review, Existing Systems Evaluation
   <u>Description:</u> The system should present a complete set of links for launching tutorial activities. Each link should have a picture that promotes user recognition of its function, as well as text to provide reinforcement so that the user does not have to recall the method necessary to navigate to the task they desire.
   In consideration of the user's potential lack of accuracy over the screen pointer (due to their control over the mouse), the link should also form a large clickable-area. They should also be well spread out across the interface to prevent the user from launching an activity accidentally.

2. The system should present tasks using appropriate learning metaphors.

Origin: Literature Review, Stakeholder-Centred
Description: Tasks should use appropriate learning metaphors so that they are readily accessible for child users. Learning metaphors should help to bridge the gap between the abstract mathematical concepts and concrete applications and thus encourage children to explore applying classroom taught concepts in interesting situations

3. The system should provide a set of tasks that 'change' to support the user's educational development.
   Origin: Literature Review, Stakeholder-Centred
   Description: The set of tasks should be *different* every time they are performed. The questions posed and data used to create mathematical scenarios should differ so that user is required to apply their knowledge and not just recall answers from memory.
   The user should also be able to apply their knowledge in more difficult scenarios so to nurture and expand their level of understanding of the mathematical concept at hand.
   User related modification: The tasks should not increase in difficulty where the user is unable to control this change. The user should be able to specify (using levels for example) if they want to apply to attempt more difficult tasks.

4. The system should use graphics and animation to enable the presentation of highly visual tasks for the children to apply their knowledge within.
   Origin: Literature Review, Stakeholder-Centred
   Description: As discussed in the literature review, foundation stage children are dominated by their sense of visual perception, and accordingly all tasks should be highly visual so that they are easy for children to interpret, provided they are accompanied by appropriate auditory stimuli. Also graphics intensive tasks should by their nature be engaging and fun for child users, and thus help to maintain an efficient learning session.

5. The system should use speech and sound output as the primary mode of communication with the user.
   Origin: Stakeholder-Centred
   Description: Foundation stage children have a limited ability to decode text and can more readily respond to speech and sound. Audible communication mirrors the teacher-child interaction as seen in the classroom, and so should provide the user with some level of familiarity when using the system. It also provides the opportunity for children to practice listening to and interpreting instructions where mathematics vocabulary is used in appropriate contexts.

6. The user should be able to repeat all audible instructions and feedback.
   Origin: Stakeholder-Centred, Existing Systems Evaluation
   Description: Given that speech and sound are by their nature non-permanent, the user should be able to repeat the instruction set if they missed part of it or are confused by the message it conveys.

7. The system should use an interface persona to direct and manage the task set.
   Origin: Literature Review, Stakeholder-Centred, Existing Systems Evaluation
   Description: A persona should be used as the source for communicating audible instructions and feedback. The persona should not be intrusive to the functionality of the system or inhibit the child's natural thought processes.
   In relation to Nielsen's usability principle stating the need for a suitable match between system and the real world, the interface persona should "speak the user's language". It should introduce the task by providing a brief overview of how the user should perform the task (including the aim) and how they must enter their answer.

8. The system's mode of communication with the user should have a 'permanent' nature.
   Origin: Literature Review, Existing Systems Evaluation

Description: Further to the audible communication given by the interface persona, key words should be printed to screen via text to present the salient points of the instruction or feedback. This should act as a reference for the user on the aim of the task or the main message of the feedback.

9. The system should provide appropriate feedback to the user.
   Origin: Literature Review, Existing Systems Evaluation
   Description: This relates to Dix et al's design principle of system responsiveness. The tutor should provide feedback to denote whether the user has answered a question rightly or wrongly, allowing them to gauge the level of their achievement.
   If a correct answer is given, feedback should be positive and encouraging so to enhance the user's satisfaction from completing the task. As Keller (1983) states feedback of this nature positively affects a user's level of motivation.
   If a wrong answer is given, quantitative feedback (for example "too few" or "too many") should be given to providing the basis for the user to recognise and possibly reform their mistake. Quantitative feedback should be followed by the system allowing the user to have a second chance at performing the question before revealing the correct answer if they answer wrongly.

10. The system should allow the user to control all tasks.
    Origin: Literature Review, Existing Systems Evaluation
    Description: This requirement relates to Nielsen's usability principle of user control and freedom.
    The user should also not be constrained to a strict procedure for completing the task without the option to quit a task at any time.
    The user should not be 'locked-out' of the interface for prolonged periods whilst instructions are being given or demonstrations are being enacted.

11. The system should allow the user freedom to self direct the learning session
    Origin: Literature Review, Existing Systems Evaluation
    Description: As promoted by the DFEE (1999), users should have the freedom to explore the interface and its tasks, thus enabling the benefits of ICT media in a learning context.
    The user should be able to control the learning session by customising its level of difficulty to some degree, in pursuit of the development and extension of their knowledge base (relating to requirement 3). We must consider that any incremental increase in difficulty should be carefully mapped to the child's mathematics ability and knowledge from classroom activities.

12. The system should have a reward scheme, providing a goal for the learning session.
    Origin: Stakeholder-Centred, Existing Systems Evaluation
    Description: Further to system feedback, the reward scheme should give the child a local goal for the learning session that they strive to achieve. The reward scheme should convey the user's achievements from completing the task.
    The reward scheme should also allow visibility of system status (a usability principle stated by Nielsen, 2001), such that the user can see 'where they are' in the task and thus assess their progress towards the reward scheme goal.

13. The system should provide a printable summary of the questions answered.
    Origin: Stakeholder-Centred, Existing Systems Evaluation
    Description: Both the questions answered and the user's relative performance in the task should be summarised by the system, allowing a hardcopy to be printed for the purposes of the foundation stage profile.

14. The system should provide a set of teacher controls to allow appropriate customisation of the system interface and the tutorial content.
    Origin: Stakeholder-Centred, Existing Systems Evaluation
    Description: The teacher should be able to access a set of controls allowing them to customise the interface and the tutorial content. These controls should only be available to the teacher and not child users. The exit from the system should also be located in this area so that child users cannot accidentally exit the system during a learning session.
    User related modification: This is a low priority requirement, as Louise Aylott stated that she has never used the customisability settings for any other tutoring system in the classroom. She did however emphasise the need for the exit button (from the system) to be located away from the children's activities and tasks to prevent accidental error.

2.4 Non-functional requirements

These requirements capture the constraints on the system's performance and its functionality, as well as facts and assumptions relevant to the system:

15. The system should be an effective teaching tool allowing children to practice applying mathematical concepts taught in the foundation stage classroom.
    Origin: Literature Review, Stakeholder-Centred
    Description: The mathematical content of the tasks presented by the system should be clearly mapped to the foundation stage national curriculum as well as classroom lessons so that concepts are not introduced above the child's schema.

16. The lesson length should be approximately 20 minutes.
    Origin: Stakeholder-Centred
    Description: The tutor should provide the teacher with an additional teaching tool that can be used as an activity from the 'job box'. In order to comply with other job box activities the lesson length should be approximately 20 minutes.

17. The system should be reliable and not produce any unexpected behaviour.
    Origin: Literature Review
    Description: The interface should comply with Nielsen's usability principle of error prevention. Thus comparable actions should have predictable results so to avoid unexpected system behaviour and user surprise. Any exceptions generated by the system should be handled in a way such that the system does not fail.

18. The system must be able to run on the on the classroom computers.
    Origin: Stakeholder-Centred
    Description: The system must comply with the following set of hardware requirements, sourced from the computing hardware available to the children at North Baddesley Infant School. We have to assume that this is representative of the computing hardware that is available to infant schools in general.
    - Microsoft Windows 1998
    - Pentium 500Mhz processor
    - 64Mb RAM
    - An SVGA display capable of supporting 16-bit colour depth and a screen resolution of 800x600 pixels
    - A graphics card capable of supporting 16-bit colour depth at 800x600 pixels
    - A Sound card and a pair of headphones

    User related modification: The use of the headphones is essential when the system is used in the classroom, so not to distract other children.

19. The system should be capable of running as a standalone program.

Origin: Existing Systems Evaluation
Description: A more general usability aspect gathered from evaluation was that the user should not have to download additional software to use or view any part of the system, as this requires an assumption to be made over the user's internet connection.

2.5 Usability and user interface requirements

These requirements capture the important usability goals and associated measures for the system:

20. The system should instantiate a graphical direct manipulation interaction style.
Origin: Literature Review
Description: An appropriate combination of point-and-click and WIMP interaction styles should be used to promote a high level of user interaction in an intuitive interactive environment. This is the most appropriate interaction style given the computer literacy, the computer-interaction experience of foundation children and the nature of the problem domain.

21. The interface should allow direct manipulation of objects for relevant tasks, so to provide a different perspective on how to arrive at an answer in a mathematics context.
Origin: Literature Review
Description: Direct manipulation of objects provides the means for highly interactive tasks where the user constantly inputs instructions and receives rapid feedback on the status of those instructions. Where the user can apply actions directly to objects in a highly responsive interface they can also learn to exert more control over them, which Newton (2001) states as important if a system is to engage a child in a learning session.
Direct manipulation of objects also supports the previous requirement of providing visually-rich tasks (requirement 4).

22. The user interface should be easy to understand, operate and navigate.
Origin: Literature Review, Existing Systems Evaluation
Description: Users should be able to apply 'obvious' actions on objects such that their natural thought processes are easily translated into input at the interface. Thus the interface should promote recognition of how to perform actions rather than requiring the user to recall information from memory.
The system's navigational structure should be predicable and so that navigational actions do not cause surprise to the user. The user should not be constrained from exploring the interface or the task and thus buttons should be provided to support their needs.
Accordingly principles governing usability, HCI and design should be applied and adhered to so to create a highly usable interface that elicits positive responses from the user.

23. The user should not be able to click any buttons or hyperlinks outside the scope of the tutorial area.
Origin: Existing Systems Evaluation
Description: The user should not be able to navigate away from the system and thus become 'lost'. With the limited experience foundation stage children have of computer systems this is especially salient as they do not have the skills to navigate unfamiliar systems unaided, and thus undo accidental navigational actions.

24. The interface should be highly learnable, allowing users to easily pick up how to use the system.
Origin: Literature Review, Existing Systems Evaluation

Description: The system should have a consistent colour scheme where colour pairings are carefully considered so that all text and pictures are easily legible. As seen in all evaluated systems, a consistent background colour should also be used throughout the interface to denote that the user is in the 'tutorial area'.
The same font should be used for all text in the system, to avoid possible confusion.
Further to this the interface should have consistently placed buttons for common actions (exit for example) and the persona should also operate in the same area of the interface, so to adhere to the predictability and familiarity design principles as stated by Dix et al (2000).

25. The system should allow a child to work alone, without the need for teacher guidance.
Origin: Stakeholder-Centred
Description: Given the role as an activity from the 'job box', the tutor should allow a child to work alone, or in collaboration with other children, without the need for teacher guidance or support. The system should be suitably intuitive so that a child can interact with it and perform all the actions they desire without the need for third party interpretation or explanation.

26. The system should be accessible to more than one user at a time.
Origin: Literature Review
Description: Children at the foundation stage are encouraged to develop shared understandings with other children and adults, as it is proven that they learn effectively in collaboration. Thus the system should be accessible to multiple users allowing them to discuss tasks.
This requirement is enforced by the fact that there is an occasional need for sharing of computers in the classroom.

27. The system should be able to be used in conjunction with the other computer-based tutors used in the classroom.
Origin: Stakeholder-Centred
Description: The system should complement the tasks and activities presented in other systems and thus not offer any conflicting messages that could potentially confuse child users.

2.6 User requirements

These requirements capture the characteristics, abilities and skills of the intended user group:

28. The system should support the user's level of mathematics competency.
Origin: Literature Review
Description: The mathematical content of the system should be clearly mapped to the foundation stage national curriculum. It should not introduce new topics or present scenarios that are inappropriate given the mathematical abilities of foundation stage children.

29. The mouse should be used as the key input device for interacting with the computer.
Origin: Stakeholder-Centred
Description: Child users can comfortably operate the mouse as an input device. Therefore appropriate point-and-click, drag-and-drop and selecting operations (direct manipulation) should allow users to perform all required actions and navigate the interface. This requirement relates to the system's usability as well as being a user requirement.

30. The interface should allow a large margin for error to compensate for the user's potentially limited control over the mouse and thus the screen pointer.
    Origin: Literature Review, Stakeholder-Centred
    Description: The interface should have appropriately sized buttons that are well spaced to compensate for child user's limited control over the mouse as an input device. Buffer areas should be used to compensate for mistakes to minimise the chance of a user initiating an accidental action.

31. The system should not require the user learn any new skills, and thus it should be tailored to their skill set and computing experience.
    Origin: Stakeholder-Centred
    Description: Through discussion and observation is emerged that the keyboard is not an appropriate input device due to the children's limited skills over its operation.
    Child users do not have any experience of issues over multitasking and thus activities that require switching windows and simultaneous thought processes are not appropriate.
    Child users do not have any notable experience of using scroll bars to focus on key parts of the interface as all packages used in the classroom require the children to perform tasks in a fixed task domain. Therefore the interface should not require the user to scroll to different parts of the interface.
    Child users do not have any experience of initiating a learning session with the computer. Consequently the teacher must load the software prior to it being used by the children.

Design Appendix

1. Evaluation of Existing Tutoring Systems
2. Low-fidelity Prototype
3. Redesign of Low-fidelity Prototype
4. Design Solutions

1 Low-fidelity Prototype

This chapter includes the essential details of each screen of the interface, as well as its navigational structure. For the purposes of the user-centred discussion and redesign, only the screen mock-ups were presented to the user, and then explanation and questioning from the author was used to probe salient aspects of the design to gauge the user's opinion.

1.1 All Screens

In the pursuit of developing a highly usable system, it is necessary to design a number of common features of the user interface to promote the design heuristic of learnability (Dix et al, 2000). Learnable systems allow novice users to easily understand how to pick-up and perform actions relating to their tasks, and also how to navigate the interface without the need for assistance or instruction. Complying with this design principle is vital if the system is to be truly usable in the classroom environment given the skills and computing experience of foundation stage children, as clearly they must be able to access the functionality of the system without being constrained by the short-comings of the user interface.
The importance of learnability was stated as usability requirement 24 for the system, with a clear relation to usability requirement 22, such that the system should be easy to understand, operate and navigate. Accordingly we must design a common set of features that each screen of the interface will have:

1.1.1 Screen Size

The interface should fill the computer screen so that child users cannot click on buttons or icons outside the scope of the system (usability requirement 23). Therefore the interface must run at 800x600 pixels as dictated by the computing hardware used in the foundation stage classroom.

1.1.2 Interface Frame

An interface frame will be used to enclose the interface and give each screen a feeling of familiarity and consistency. The frame will have the same colour and structure for each screen.
The interface persona will be placed *within* the interface frame so to separate it from the task domain as well as the rest of the interface. As stated in functional requirement 7 the persona should direct and manage the task set by conveying all audible instructions and feedback, but it does not have any active part in the task itself. Therefore enclosing the persona in its own frame denotes that the user cannot interact with it and thus prevents any potential confusion.
In relation to functional requirement 6, a button should be provided that allows the user to repeat any audible communication on demand. This was recognised as a well designed feature of the Tizzy's Toybox tutor and thus has been applied to the system design. The repeat button should be located in the frame with the interface persona to denote their relation, and also to separate it from other navigational buttons in the interface.
Furthermore, functional requirement 8 states that the system's mode of communication should have a permanent nature, and thus a box for printing key words to screen should be located below the interface persona, providing a reference on the key points of instructions and feedback. Being located with the persona the user only has to reference one area of the interface to interpret instructions or feedback, providing a consistent source for communication.

1.1.3 Fonts

The fonts used in the interface for similar purposes should be the same size and type throughout. Hence all text printed to the key-word-box should be the same size and type, as should the text labels for all buttons. Importantly all text used in the interface should be lower case as this is more fully understood by foundation stage children (as directed by foundation stage teacher, Louise Aylott). Uppercase letters of the alphabet are not lessons of the foundation stage year and thus they cannot be used with the expectation of being interpreted correctly.

1.1.4 Buttons

All navigational buttons should be named considerately so to promote recognition of their action. Where buttons perform the same navigational action they should be placed in the same location in each screen and have the same name. This promotes predictability and synthesisability (Dix et al, 2000) such that the user can apply their knowledge of previous interactions in order to predict the behaviour of buttons throughout the interface.

All buttons in the interface should perform consistent effects to denote whether they are up, down, rolled-over or hit:

- Up: As already stated, all buttons should have a clear label and picture if appropriate. This should clearly define their function, with the aim of promoting user recognition of the action they represent.
- Roll-over: All buttons should provide feedback to the user to denote that the screen pointer is 'over' it. In consideration of children's potentially limited control over the mouse, it is important that they are fully aware that they are 'on' the button hotspot. When evaluating Teddy Bear's Picnic it was noted that audible feedback was given to articulate the function of the button that the pointer was over. However, this could cause confusion if a user rolls-over a number of buttons quickly. Therefore all buttons should simply highlight their text label when the user is 'over' them.
- Down: All buttons should provide feedback to denote that they are 'down', so to provide reinforcement that the user is performing an action on them. Therefore they should change colour whilst in the 'down' position, but subsequently return to their 'up' state.
- Hit: The button's action (whether navigational or otherwise) should be performed immediately after they have been hit, so to prevent user confusion where their action is delayed.

1.1.5 Background Colour

The same background colour should be used for related screens in the interface so to reinforce their relationship; the same colour background should be used for all task screens, a different colour should be used for all teacher controls screens, and another different colour should be used for the main menu. Therefore any change of scope in the interface where the user navigates to a different zone of interaction (for example navigating from the main menu to the task) is denoted by a change of background colour. This defines the fact that there is a clear difference between the zones as well as providing some reassurance to the user that they are no longer in the scope of the previous activity.

Given the specification of the features common to all screens in the interface, we deal with the design of individual screens in the following sections.

1.2 Main Menu

There are two figures for the main menu of the interface, relating to the teacher control option for changing the appearance of the main menu (refer to section 1.4). Figure 1a denotes the main menu, where the user must perform the task in a temporal fashion. Figure 1b denotes the main menu where the child can attempt any level at any time.

Figure 1a: Main Menu Screen (temporal)

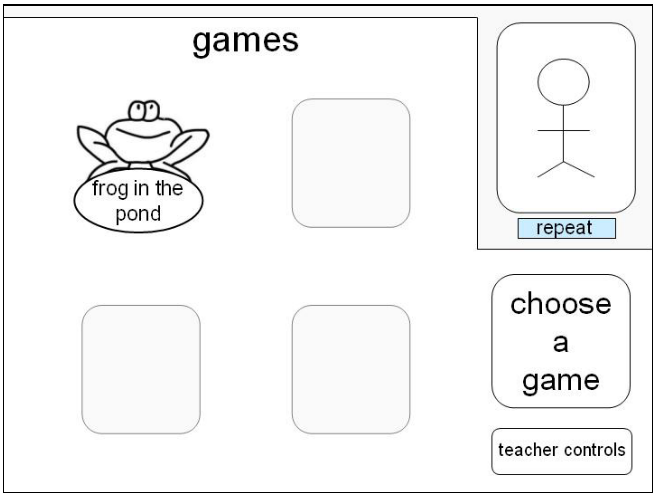


Figure 1 denotes the main menu for the interface. ‘Games’ is used as the title here rather than ‘main menu’, as main menu would carry little meaning for the children and could potentially be interpreted incorrectly. Also the title ‘main menu’ is normally used to signify that a screen is the centre-point of an interface, but the children would not necessary pick up on this, and thus its use is redundant.
Denoting the task as a ‘game’ is also more closely related to classroom terminology, where the words game or job are used for describing educational activities. Moreover a child would prefer to embark on a game rather than a ‘learning task’ due to its novelty appeal.

Colour: The main menu should have a background colour that is different from all other parts of the interface to denote that it has a different function.

Navigation: The iconic button to the ‘frog in the pond’ task is large in size and it is also located away from other buttons both to compensate for children’s potentially poor control over the mouse and to minimise the chance of launching an unwanted navigational action (relating to user requirement 30). The button also uses the frog and lily pad icon to promote recognition of the task it represents.
The button to the teacher controls is located away from the frog in the pond iconic button to denote that is separate from the “child’s” area of the screen. It is located in the same place as the ‘return to main menu’ buttons located on the task screens (section 1.3), and the teacher controls screen (section 1.4) to provide some consistency in the placing of navigational buttons that move out of the scope of the current activity.
The three ‘fake’ buttons in this screen denote areas where iconic buttons to other lessons can be added if the system is to be extended in the future.

Information: The interface persona gives the audible instruction that the user should choose a game, which is reinforced by the words printed to the key-word-box to provide some permanence to this instruction (relating to functional requirement 8). This audible instruction can be repeated on demand by clicking the repeat button, which is located in the same frame as the persona to denote their relation.

Figure 1b: Main Menu Screen (options)

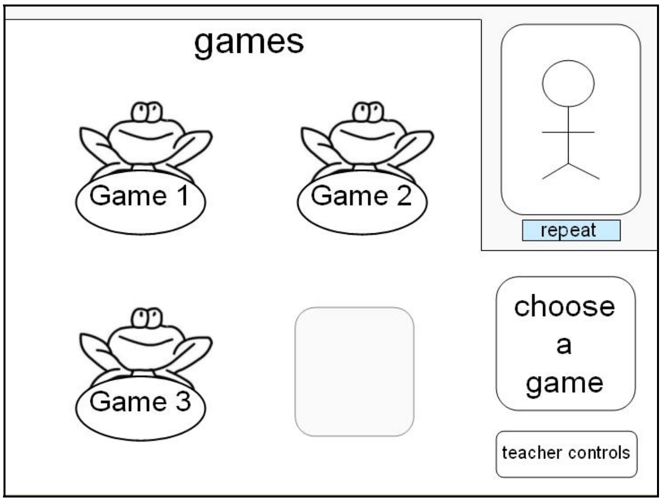


Figure 1b denotes the main menu screen for the 'options' structure. It will be identical to that given in the description for figure 1a, the only difference being to the navigational structure.
There are three iconic buttons to each level of the 'frog in the pond' task that are large in size and appropriately spaced across the screen to compensate for the aforementioned potentially poor control over the mouse. The button uses the frog and lily pad icon and a text label to promote recognition of the task level it represents. Clearly "game 1" represents level 1, "game 2" represents level 2, and so on, which should allow the user to appreciate the sequence of the tasks as well as the fact that the higher the number, the harder to task.

1.3 Task – figures 2 – 7

The format of the actual task and thus the task screens is exactly the same for both the temporal structure of the main menu and the options structure. As already stated the difference between the two structures is that the temporal structure dictates that the users can only access level 2 after performing level 1, and can only access level 3 after performing level 2, whereas the options structure allows the user to access any level of the task at any time. In accordance with this, the screen that is displayed at the end of the task (after the reward scheme) that requires the user to decide what they would like to do next, must be different between the structures, as shown in figures 7a and 7b.

Figure 2: Task Screen

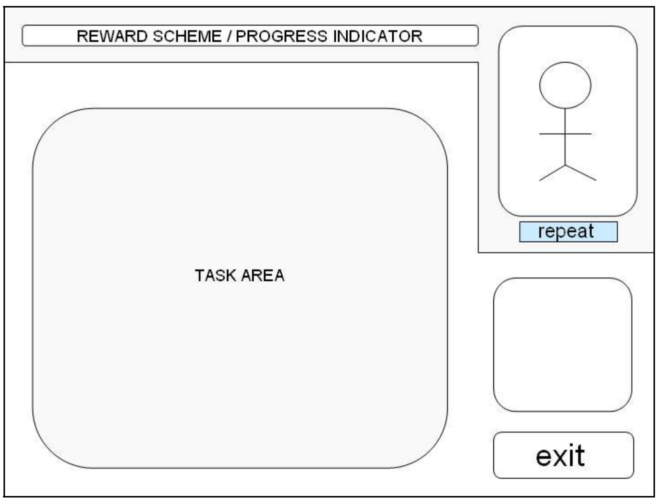
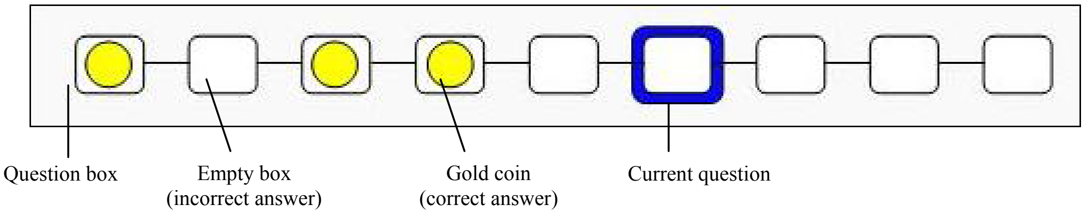


Figure 2 denotes the task frame. This is consistent for all task related screens.

The task area is *where* all interactive activities relating to the task occur, including where the user must enter their answer. Having a definite area for this ensures that the user cannot become confused by having to interact with any other parts of the screen. It also reinforces the fact that the interface persona, the key-word-box and the reward scheme do not form 'part' of the task and cannot be interacted with.
The 'reward scheme bar' is displayed at the top of the screen. As previously stated, the user accumulates gold coins for each question answered correctly. There are 8 spaces for coins in this bar representing each question of the task and thus when the user answers a question correctly a gold coin is placed in its relevant box (refer to figure 2a). When an incorrect answer is given, no gold coin is awarded and the question box is left empty. Moreover, the current question-box is highlighted to making it possible for the user to adjudge their progress through the task (relating to functional requirement 12).

Figure 2a: Reward scheme area and progress indicator



The reward scheme and progress indicator have been designed in consideration of the need to comply with Dix et al's design principle of system observability (2000) as well as Nielsen's usability principle of visibility of system status (2001). Based on the status of this indicator the user is able to evaluate the current stage of the task; how many questions they have left until it is completed and also how many they have attempted (in figure 2a the user can recognise that they have answered 3 questions correctly and have 3 to go). The progress indicator is an important part of the interface as it reduces the propensity of the system to 'trap' the user in a task where they do not have appropriate information to adjudge 'where they are'.

Colour: The background colour should be the same for all task screens to denote that the user is 'in the tutorial area'. If the user were to subsequently move away from the task (for example by using the exit button to quit the task and return to the main menu) this will be made explicit by the change of background colour thus denoting that they are no longer in the scope of the task.

Navigation: The exit button that returns the scope to the main menu is located away from the task area to denote that is separate from any part of the task so to ensure it is not clicked by mistake. On all task screens this button is located in the same place and has the same label.

Information: The persona delivers all audible instructions and feedback related to the task (relating to functional requirement 7). At the start of each level of the task the persona states the aim and also describes how the user must input their answer. This can be repeated on demand by clicking the repeat button. The key points of the task are printed to the key-word-box to provide a reference to the user on what they must do.

Figure 3: Task Level 1

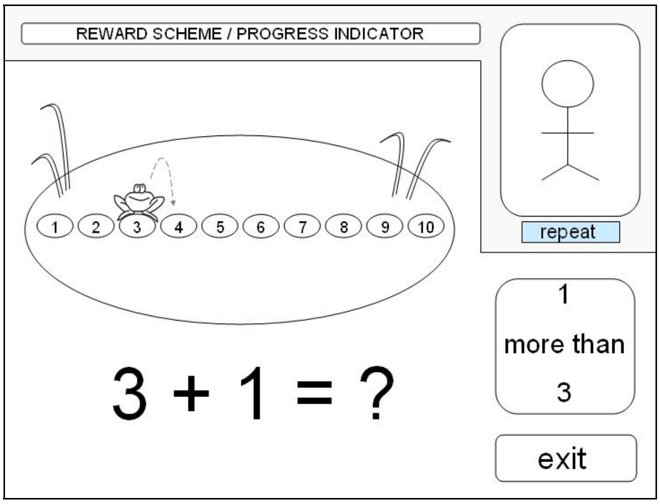


Figure 3 denotes level 1 of the task. This level provides a lesson that is designed to mirror the 'number in the bag' lesson used in the foundation stage classroom. It also provides the opportunity for children to practice interpreting and recognising numbers of the number sequence as well as listening to correctly articulated mathematical language.

The key point of using the 'frog in the pond' learning metaphor here is that the lily pads provide a set of objects that the user can count in order to arrive at their answer. By jumping between the lily pads the frog also provides the means by which to demonstrate how the child should count *up* the number sequence to find the number that is 1 more than 3 in the example shown. The need to use graphics intensive, highly visual task presentations was cited as functional requirement 4.

Information: The language used for instructions and feedback should be closely mapped to the 'number in the bag' game that was observed in the classroom. By replicating this language we can recognise that the children will be familiar it, thus making the task more intuitive and also meaning that it is closely mapped to the mathematic language promoted by the national curriculum. Accordingly the following format should be used for audible communication:

- Instruction: "What is 1 more than 3?" At this point of the question the frog is positioned on the lily pad that relates to the 'starting point' in the number sequence from which the user must calculate their answer (i.e. 3).
- Feedback (if correct): "Yes that's right! 1 more than 3 is 4". Subsequently the frog jumps to the appropriate lily pad to reaffirm the correct answer and thus consolidate the child's learning. As part of the reward scheme, the user is awarded a gold coin for this correct answer.
- Feedback (if wrong initially): The initial feedback to a wrong answer should be quantitative so that the user is given guidance on where they went wrong, potentially enabling them to correct their mistake. Quantitative feedback was stated as an important in functional requirement 9, as it gives some direction to the child when they attempt the question for a second time. Therefore the phrases "no, that is too few" or "no, that is too many" should be used. Subsequently if the user gets the question right they are rewarded with a gold coin.
- Feedback (if wrong for second time): If the user gives the wrong answer after a try again, the correct answer should be revealed, potentially enabling the user to recognise their mistake. For example "no, actually, 1 more than 3 is 4." Subsequently the frog should jump to the appropriate lily pad to demonstrate the correct answer. No gold coin is awarded for a second incorrect answer.

All audible instructions and feedback can be repeated on demand by clicking the repeat button. The key-word-box provides a reference for the key point of the question, and thus changes for each question posed.

Interaction: The user is required to use a point-and-click operation to select the lily pad in the sequence that relates to their answer (from the set of lily pads, hence multiple choice). Therefore in the example "1 more than 3", the user must click on lily pad 4. This relates to usability requirement 20 such that the task should instantiate a direct manipulation interaction style and use appropriate mouse directed methods for input. This mode of input is used in all subsequent levels of the task providing consistency and generalisability (design principles, Dix et al 2000) such that the user can apply their knowledge of interacting with the system in all levels of the task.

Navigation: After the correct answer has been demonstrated, a 'next' button appears in place of the equation (at the bottom of the screen), which the user must click in order to move onto the next question. This allows the user to move through the task at their own pace and thus self-direct their learning session (functional requirement 11).

Figure 4: Task Level 2

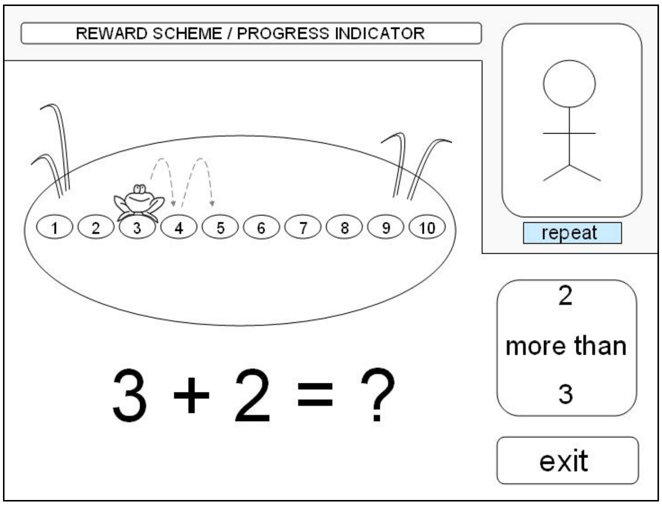


Figure 4 denotes level 3 of the task. This level provides a lesson that extends the previous one by providing the opportunity to apply their knowledge of the number sequence in a more difficult context. Similarly to level 1, this level also reflects the 'number in the bag' lesson that was observed in the classroom and thus previous discussions relating to level 1 are relevant here also, with the exception of a small change to the information that the screen displays:

- Instruction: "What is 2 more than 3?"
- Feedback (if correct): "Yes that's right! 2 more than 3 is 5". Subsequently the frog should jump to the appropriate lily pad to reaffirm the correct answer.
- Feedback (if wrong initially): This is exactly the same as the previous level meaning that the user can predict and recognise the system feedback.
- Feedback (if wrong for second time): If the user gives the wrong answer after a try again,, the correct answer should be revealed; for example "no, actually, 2 more than 3 is 5." Subsequently the frog should jump two lily pads up the sequence to demonstrate the correct answer.

Due to the similarity of this level to level 1 we can recognise that the system is highly learnable due to the familiarity and generalisability of the mode of interaction, the screen presentation and the information provided.

Figure 5: Task Level 3

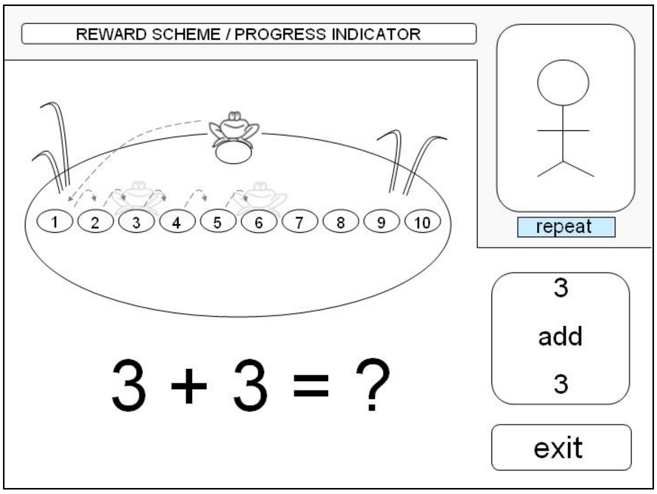


Figure 5 denotes level 3 of the task. This level provides a lesson that is designed to mirror the 'introduction to addition' lesson that was observed in the foundation stage classroom. It also provides the opportunity for children to practice listening to correctly articulated mathematical language as applied to equations over addition.

Similarly to the previous two levels the frog in the pond learning metaphor is important here as the lily pads provide a set of objects that the user can count in order to arrive at their answer. Also the frog can be used to demonstrate the answer by jumping between the lily pads, thus showing the user how to find the total of an equation (i.e. the user must count 3 objects and then count a further 3 objects in order to effectively calculate the answer to the example shown).

Information: The language used for instructions and feedback should be closely mapped to the lesson that was observed in the classroom. By replicating this language we can recognise that the children will be familiar it, thus making the task more intuitive and also meaning that it is closely mapped to the mathematics language promoted by the national curriculum. Accordingly the following format should be used for audible communication, which has been thoughtfully designed to the mirror the communication given in previous levels:

- Instruction: "What is 3 add 3?" At this point of the question the frog is positioned on a lily pad away from the number sequence allowing the user to use the sequence as an application to calculate their answer.
- Feedback (if correct): "Yes that's right! 3 add 3 equals 6". Subsequently the frog should jump up the number sequence to the appropriate lily pad to reaffirm the correct answer and thus consolidate the child's learning. The user is awarded a gold coin for this correct answer.
- Feedback (if wrong initially): The initial feedback to a wrong answer should be quantitative so that the user is given guidance on where they went wrong, potentially enabling them to correct their mistake. Therefore the phrases "no, that is too few" or "no, that is too many" should be used thus being consistent with the feedback used in previous levels of the task. Subsequently if the user gets the question right they are rewarded with a gold coin.
- Feedback (if wrong for second time): If the user gives the wrong answer after a try again, the correct answer should be revealed, potentially enabling the user to recognise their mistake, for example "no, actually, 3 add 3 equals 6." Subsequently the frog should jump up the number sequence to the appropriate lily pad to demonstrate the correct answer. No gold coin is awarded for a second incorrect answer.

All audible instructions and feedback can be repeated on demand by clicking the repeat button. The key-word-box also provides a reference for the key point of the question, and thus changes for each question posed. Importantly the key words should denote that 'adding' is the concept that must be applied when calculating the answer.

Interaction: The user is required to use a point-and-click operation to select the lily pad in the sequence that relates to their answer. Therefore in the example "3 + 3", the user must click on lily pad 6. This is consistent with the mode of input used in all previous levels.

Navigation: After the correct answer has been demonstrated, a 'next' button appears in place of the equation (at the bottom of the screen), which the user must click in order to move onto the next question.

Figure 6, 6a, 6b, 6c – Reward Scheme

Figure 6: Reward Scheme Screen

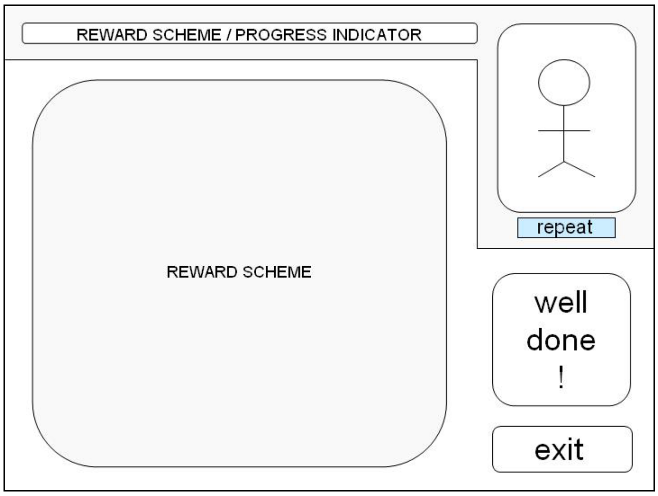


Figure 6a, 6b, 6c: Final Animation

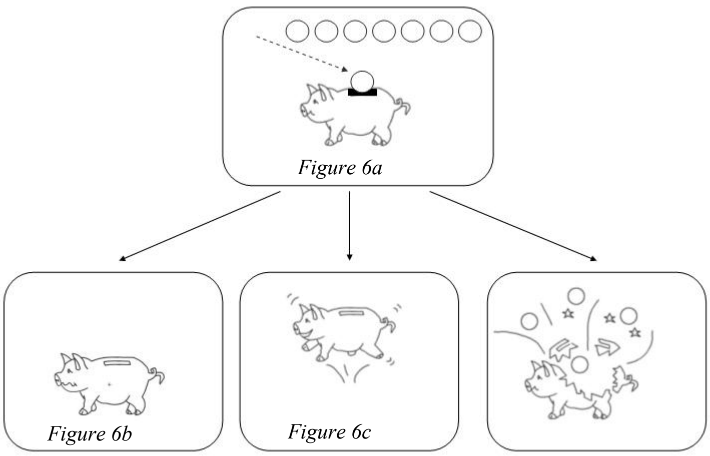


Figure 6 denotes the post-task reward scheme frame. This screen is displayed at the end of each level of the task and contains the reward scheme animation. This screen should provide some closure for the user that the level has been completed (Dix et al, 2000). It has been thoughtfully designed in consideration of the task screens so that it is consistent in terms of its layout and how it presents information to the user.

The reward scheme area is 'where' the final animation is performed to reward the child for collecting gold coins throughout the task level. As shown in figure 6a, an animation is used to demonstrate the coins entering the piggy bank, thus highlighting the relationship between the gold coins and the piggy bank animation. Given that there are 8 questions per level if the user accumulates:

- 0 coins, the piggy bank looks unhappy and walks off the screen (figure 6b)
- 1-5 coins, the piggy bank jumps up and down and then walks off the screen (figure 6c)
- 6-8 coins, the piggy bank explodes and then walks off the screen (figure 6d)

The key point of these animations is to instil some level of fun to the task, as well as rewarding the user for completing it. A more extravagant exploding animation denotes a higher achievement. The reward scheme also provides a goal for the user, such that they should strive to get more answers right in order to be rewarded with the more exciting animation.

Colour: The background colour should be the same as the task screens to denote that the user is still 'in the tutorial area' and that the reward scheme relates to the level of the task that has just been completed.

Navigation: The exit button that returns the scope to the main menu is located away from the reward scheme area to denote that is separate from the final animation. It is located in the same place as the task screens and has the same label thus providing some consistency and familiarity.

Information: The persona simply delivers audible feedback to congratulate the user on their achievements. The extravagance of this feedback is directly related to the number of coins that the user has accumulated and thus how well they have performed. Therefore "good" is fed back if the user has only accumulated only a few coins (and thus answered many questions wrongly) and "superb" is fed back if the user has accumulated a lot of coins (and thus the majority of the questions have been answered correctly).
Similarly to the task screens, this feedback can be repeated on demand by clicking the repeat button, located in the interface persona's frame. Moreover, the key point of this feedback is reflected in the key-words-box as a reference to reinforce it.

Figure 7a: End of game screen (temporal)

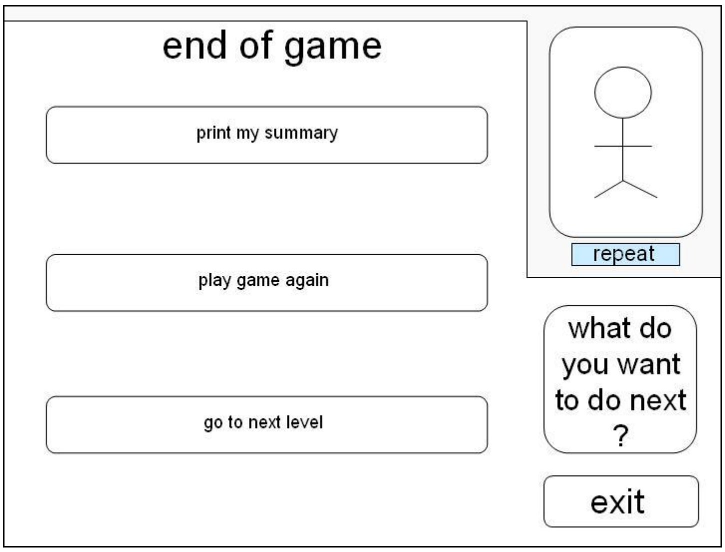


Figure 7a denotes the post-reward scheme, end of task screen that is seen if the user is performing the task in a temporal fashion. 'End of game' is used to denote that the task level is finished and the user must direct the system on what they want to do next, thus relating to functional requirement 11. The user has the option to:

- Print a summary of the task level. This relates to functional requirement 13, such that children should be able to print a summary of the task they have performed and their relative success; for the purposes of the foundation stage profile. The summary should detail the questions posed and whether the correct answer was given.
  There should not be a 'yes/no' confirmation to this action, to be consistent with comparable printing actions in the painting and word processing packages used in the classroom (thus promoting synthesisability, Dix et al, 2000). Moreover, through discussions with the teachers at the requirements analysis stage it was noted that pop-up confirmations such as this can confuse children as they are unexpected, and thus can be considered inappropriate.
- Repeat the current task level. This has been called 'play game again' and carries an arrow pointing backwards to aid its interpretability for the children.
- Go to the next task level. This enables the child to apply their knowledge in more challenging contexts and thus relates to functional requirement 3, such that the user should be able to attempt more challenging tasks if they desire. This has been called 'play next game' and carries an arrow pointing forwards to aid its interpretability for the children.
- Clearly when the user gets to the end of level 3, this button should be removed as the user is unable to progress above this level.

Importantly the reward scheme has been removed from this screen to denote that the previous task level is done. Therefore this screen provides a point of closure (usability principle, Dix et al, 2000), such that the user can prepare for the next task or end their learning session.

Colour: The background colour should be the same as the task screens to denote that the user is still 'in the tutorial area' and that they have not completed the task level as yet.

Navigation: Each button is large in size and is well spaced in the screen to compensate for children's potentially poor control over the mouse. The labels have been carefully considered so that they are easy to interpret (they "speak the user's language", Nielsen, 2001) and thus do not surprise the user when they are launched.
The exit button that returns the scope to the main menu is located away from the three main options on this screen to ensure that it is not clicked by mistake. This is consistent with all task screens to promote its familiarity.

Information: The interface persona gives the audible instruction that the user should decide what to do next, which is reinforced by the words printed to the key-word-box to provide some permanence to this instruction (relating to functional requirement 8). This audible instruction can be repeated on demand by clicking the repeat button.

Figure 7b: End of game Screen (options)

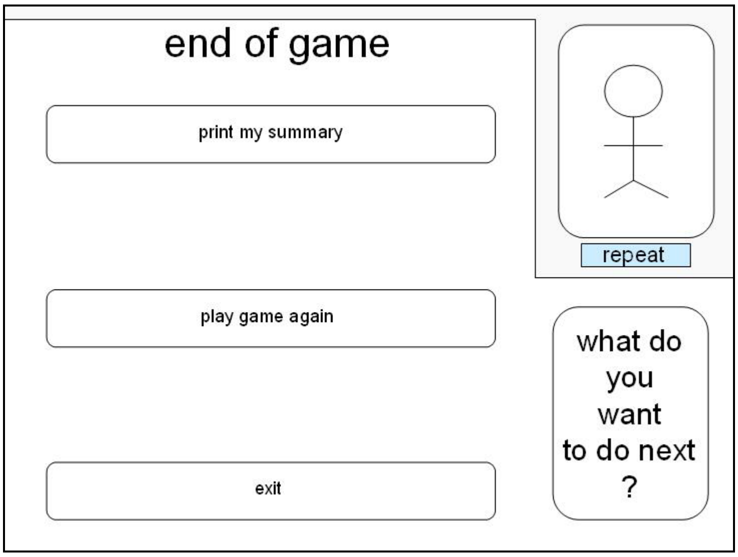


Figure 7b denotes the end of game screen for the options structure. It will be identical to that given in the description for figure 7a, the only difference being to the navigational structure. The exit button has been moved into the 'main area' of the screen to denote that it is a valid navigational action given that the user can only repeat the current level or return to the main menu at this point. Thus it is resized to be the same as the options to print the summary, or play the next level.

1.4 Teacher Controls

Figure 8 – teacher controls

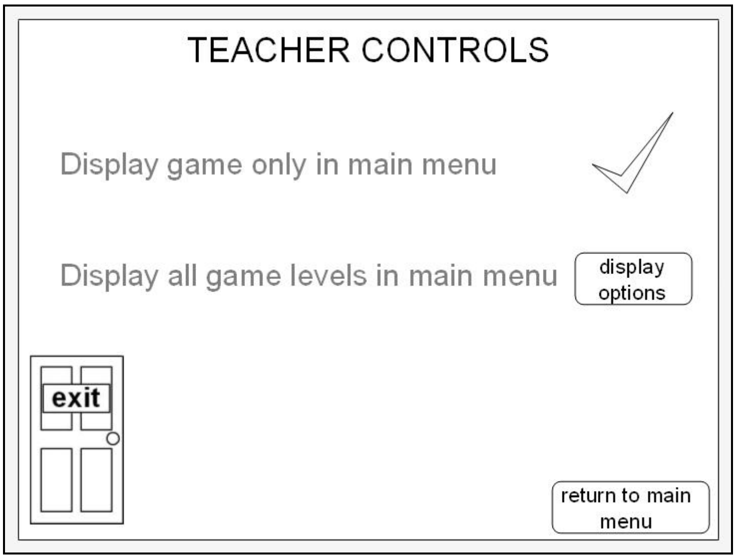


Figure 8 denotes the teacher controls screen.
As described in the Design chapter, the teacher controls has only one customisability option for the main menu. The main menu can be actively changed to either display:

- A single button to the task, where the three levels must be performed temporally.
- Three buttons, one for each level of the task, so that the user can perform them in any order, at any time.

The currently selected option will be denoted with a tick, drawing obvious parallels with classroom work in order to promote recognition for the teacher. A button is provided for the teacher to switch to the non-selected option.

Colour: This screen should have a different background colour to the main menu and the task screens, to denote that it is a different area of the interface. Therefore when the teacher navigates away from this screen it is clear that they are no longer in the scope of the teacher controls.

Navigation: The 'return to main menu' button takes the user out of the scope of the teacher controls and back to the main menu (hence its label). It is placed on the screen in the same place as the return to main menu button on the task screens thus being consistent and promoting predictability of its navigational function. This button cannot be named 'exit' similarly to the button on the task screens due to the 'exit from system' button also located on this screen; thus it is labelled 'return to main menu'.
The 'exit from system' is an iconic button and thus carries a picture of an exit door to denote that it is the 'way-out' of the system. Clicking this button launches a 'yes/no' confirmation screen (see figure 9), to ensure that the user really wants to leave the system. This means that the 'exit from system' navigational action can be un-done so to prevent the user being surprised by the system closing.

Information: It was decided that the interface persona should be removed from this screen because audible instruction is not required for the teacher.

Figure 9 – confirmation of exit

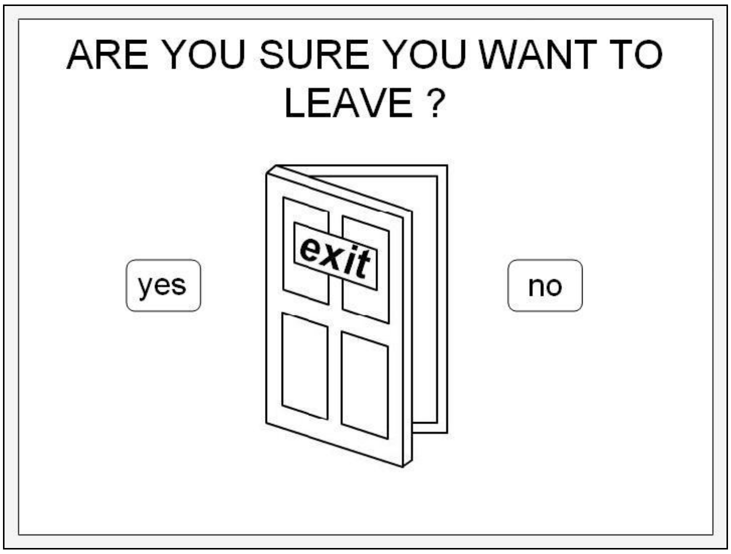


Figure 9 denotes the yes/no confirmation for closing the system. The open exit door in this screen promotes user recognition of the fact that they are about to close the interface.

Colour: This screen should have the same background colour as the teacher controls screen to denote that it is in the scope of the teacher control area of the interface.

Navigation: The buttons for user input in this screen are located on different sides of the exit door so that they can't be clicked by mistake. Clearly clicking yes closes the system whereas clicking 'no' allows the user to cancel this action and thus return to the teacher controls screen.

Information: The title makes it explicit to the user that the system will close if they click the yes button from this screen, which of course is vitally important so to prevent user surprise. Uppercase lettering has been used for this title thus going against the previous design decision to use only lowercase lettering for the purposes of easing the child's task of decoding text. However in this context uppercase lettering is appropriate as it will only be read by the teacher and is not part of the child related areas of the interface. It also denotes the importance of the message.

1.5 Navigational Structure

Having defined the essential details of each screen of the interface it is necessary to show how these screens interact and relate. This is done using an interface 'map'. Refer to figure 10 for the temporal structure, and figure 11 for the options structure.

Figure 10: Navigational structure (temporal)

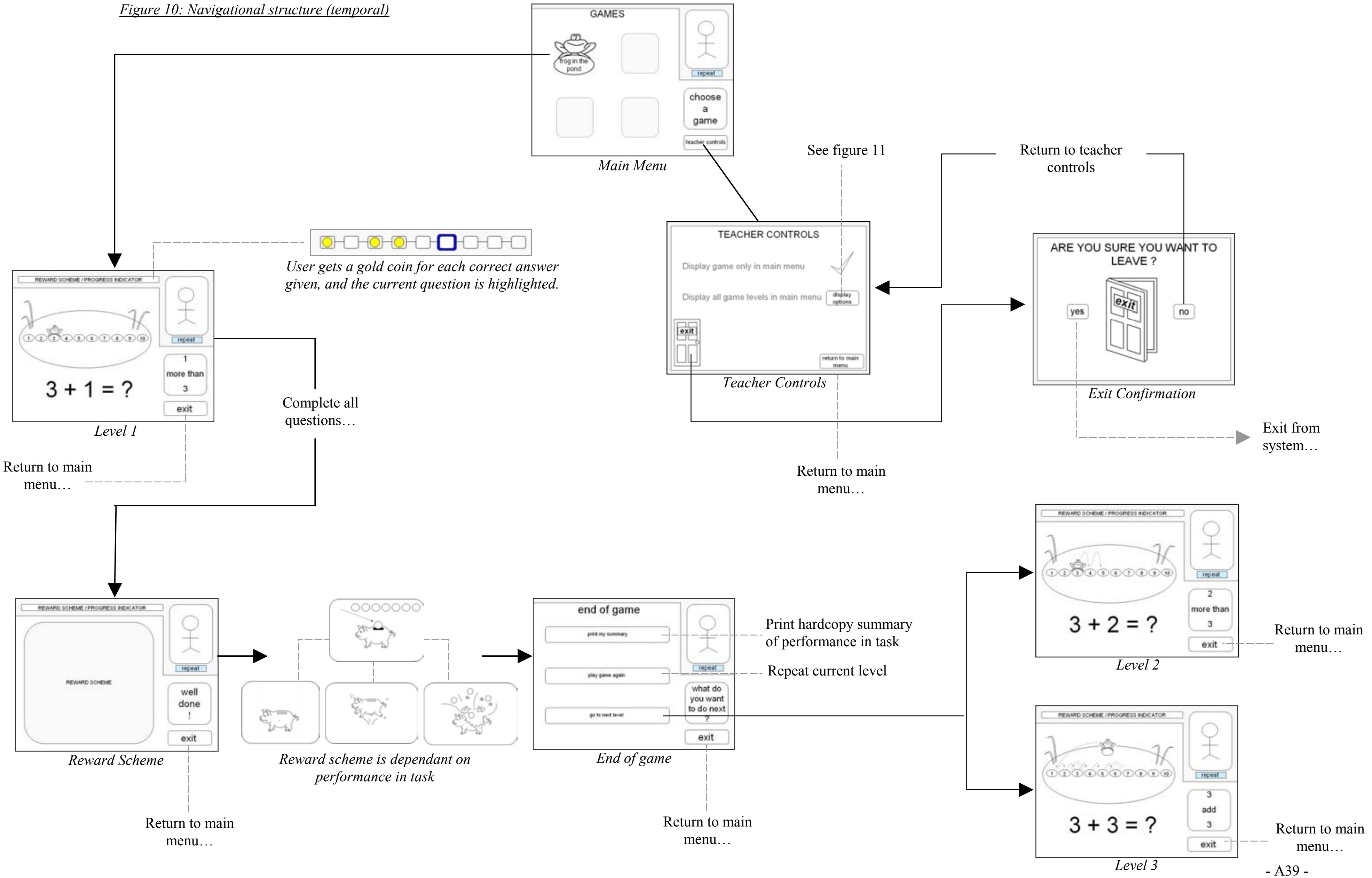


Figure 11: Navigational structure level 2 (temporal)

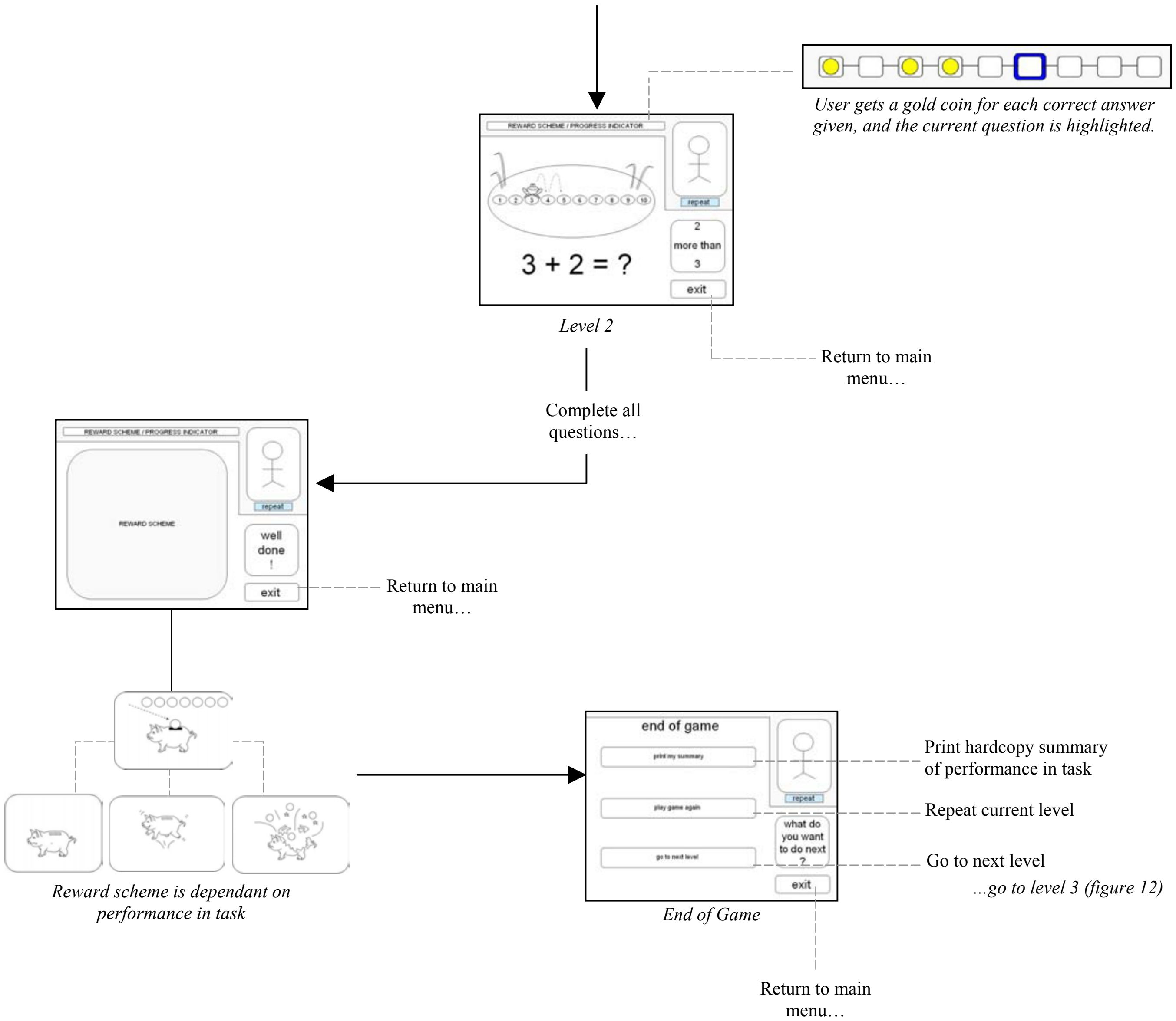


Figure 12: Navigational structure level 3 (temporal)

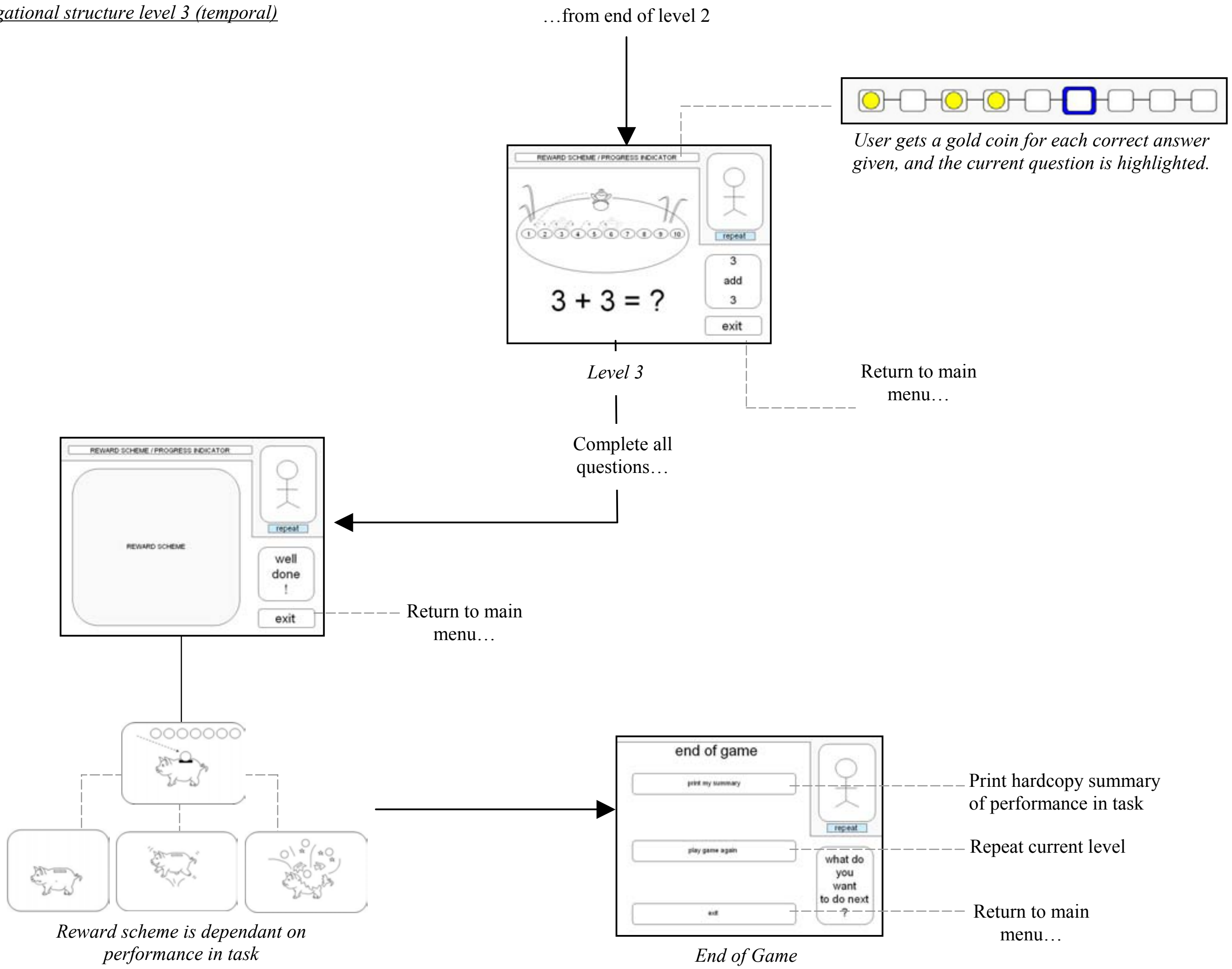
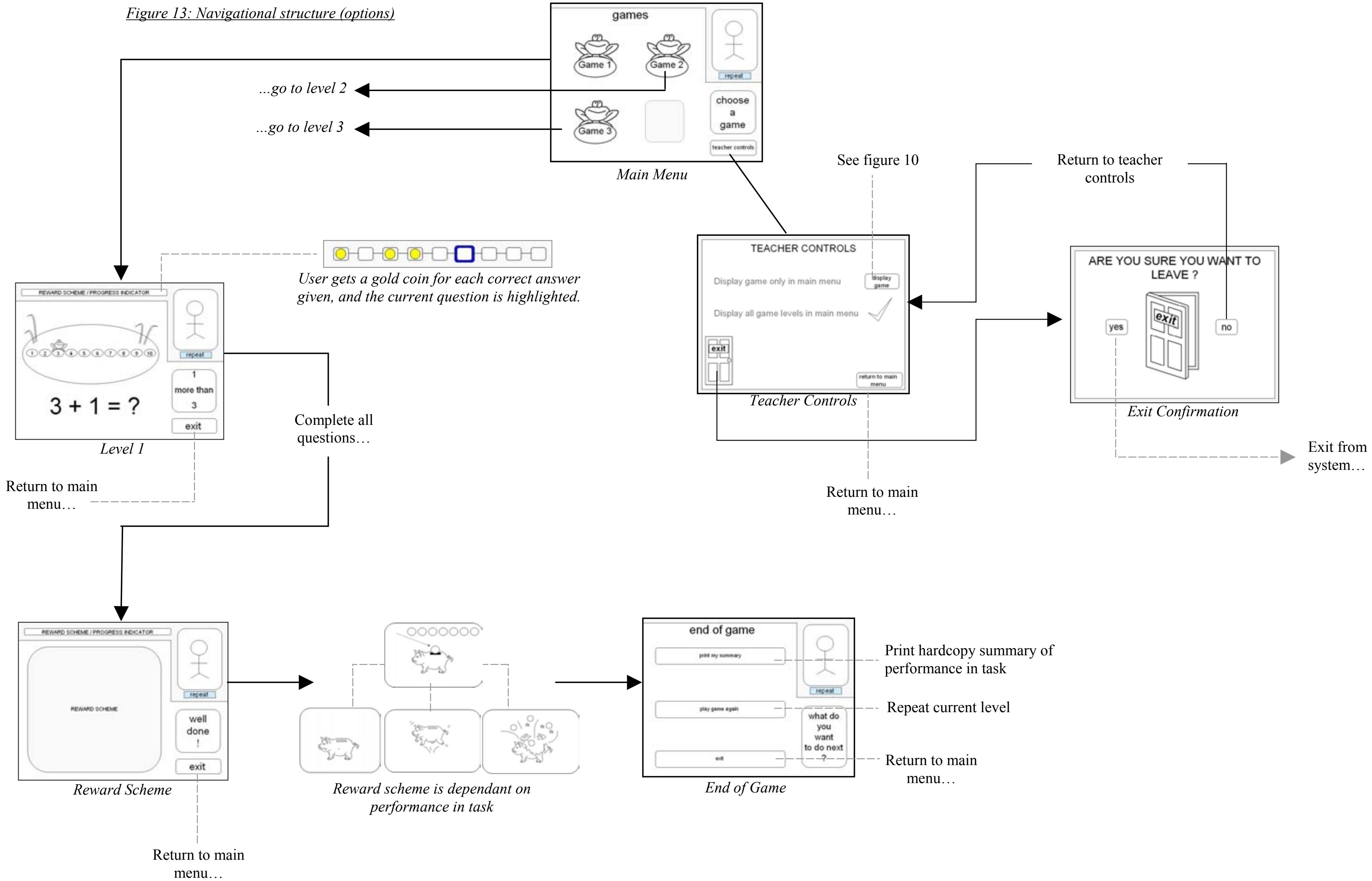


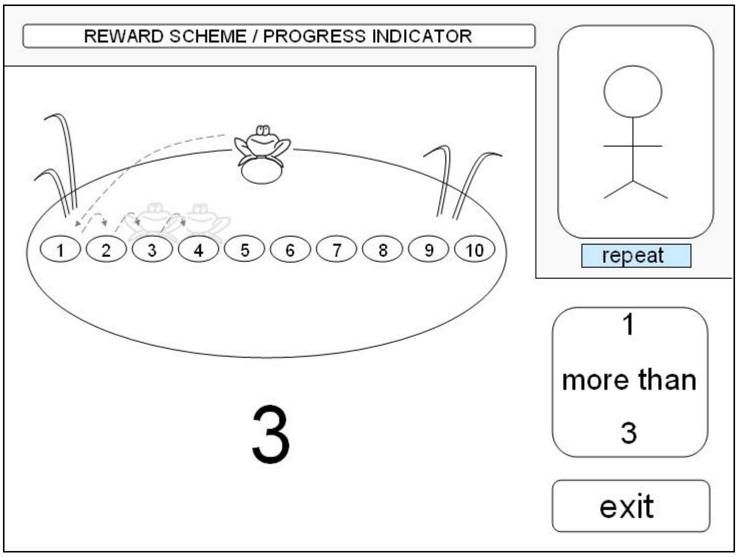
Figure 13: Navigational structure (options)



2 Redesign of Low-fidelity Prototype

The following screens denote the redesign of aspects of the initial system design, after user-centred discussion and feedback.

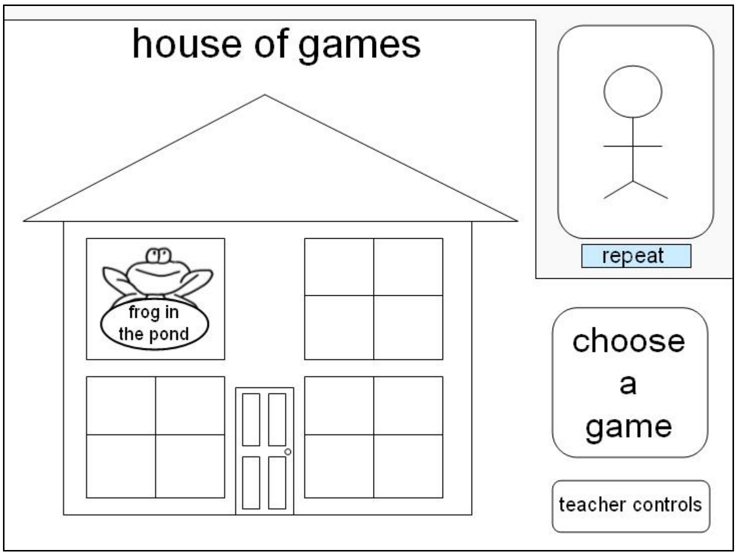
Figure 1: Redesign of task levels 1 and 2



Similarly to the previous design of levels 1 and 2, the key-word-box and the audible instructions given by the persona denote the aim of the question. As a change to the previous design, the frog is sat on a lily pad *away* from the lily pad sequence with the 'starting number' being the focus of the question (i.e. 3 in the example given). The sequence of lily pads can then be used to calculate the answer, and the user can subsequently click the appropriate lily pad to enter their answer (i.e. 4 in the example given). This more accurately mirrors the 'number in the bag' lesson of the classroom, where the children were presented with a starting number and had to use a number 'fan' as an application to arrive at their answer.
Similarly to the previous design, the frog will then jump to the appropriate lily pad to demonstrate the correct answer and thus consolidate the children's learning.

This redesign has been applied to level 2 of the task also, the only difference being that the user must find the number *two* more than the starting number (i.e. 2 more than 3 in the example above). Of course with these levels being so closely related the system promotes the design principles of familiarity and synthesisability (Dix et al, 2000) such that the user can apply the knowledge gained from interacting with the system in level one, when performing the task at level 2.

Figure 2: Redesign of Main Menu



The title has been changed to 'house of games' to mirror the redesign of the navigational buttons that return the focus to the main menu from all task screens and the teacher control screen. This promotes recognition throughout the interface.

The structure of a house has been created and there is an iconic button to the task. Clearly where the teacher selects the 'options structure' from the teacher controls screen there will be three iconic buttons on the main menu screen; one to each of the 3 levels of the task.
Using a house and locating the buttons to the game within it provides a specific location for initiating a learning 'game'. This aims to draw a parallel with the classroom, where all games are located in a cupboard and thus the children must start a game and finish a game by accessing this cupboard.

The navigational button to the teacher controls is now totally separate from the location of the buttons to the task. This reinforces the fact that the function of the teacher controls button is unrelated to these games, and should not be accessed by child users.

By redesigning the main menu we have also defined the scope for adding links to further lessons (in the other windows of the house) thus enabling future extension of the system.

3 Design Solutions

This section provides a summary of the final set of design solutions that fulfil the set of requirements and the needs and preferences of the foundation stage teachers. These solutions are presented in the same order as the requirements document, and thus are prioritised in order of importance.

3.1 Functional requirements

1. The system should display a complete set of the available task options and customisability options for the user.
   Design Solution: The system has a main menu with an iconic button as a link for launching the 'frog in the pond' lesson (denoted as a game). The main menu is named 'house of games' so to provide a 'location' for launching the task. This relates to usability requirement 22, as iconic buttons with representations of a house are used in other screens of the interface to promote recognition of the fact that the user returns to the main menu by clicking them.
   The main menu also has a link to the 'teacher controls' screen containing a set of customisability options for the interface.

2. The system should present tasks using appropriate learning metaphors.
   Design Solution: The 'frog in the pond' is used as a learning metaphor for the task as it provides a concrete scenario for the children to apply their knowledge of abstract mathematics concepts. The justification for using this metaphor is that animals are used in variety of classroom activities as props for learning and thus we can infer that the children are familiar with them in a learning context.

3. The system should provide a set of tasks that 'change' to support the user's educational development.
   Design Solution: 3 levels are used that increment in difficulty to help expand and consolidate the child's knowledge of number, the number sequence and addition. Each level can be repeated so that different questions are posed every time, meaning that static data is not simply reused.

4. The system should use graphics and animation to enable the presentation of highly visual tasks for the children to apply their knowledge within.
   Design Solution: With relation to requirement 2, the 'frog in the pond' metaphor is represented using graphics and animation so that it is easily interpretable for the children. The pond and lily pads are graphics that form the task domain (i.e. where the task oriented interaction occurs) and frog animations are used to reinforce aspects of the task instruction, and also to demonstrate the correct answer for each question. Moreover the lily pads form the mode of user input as they are the 'buttons' for the user to input their answer.

5. The system should use speech and sound output as the primary mode of communication with the user.
   Design Solution: With relation to requirement 7, the interface persona delivers all audible instructions and feedback to the user. This audible communication is closely mapped to the language used in the classroom and thus the national curriculum, as directed by foundation stage teachers.

6. The user should be able to repeat all audible instructions and feedback.

Design Solution: A repeat button is provided for the user to repeat any audible instructions or feedback on demand. This is located in the same frame as the interface persona to denote their relation.

7. The system should use an interface persona to direct and manage the task set.
Design Solution: The interface persona has been designed to be a 'teacher-style' figure to convey the importance of the information it provides, but also to provide an endearing figure for the child to relate to. The persona is located in its own frame to denote that it plays no part in the task and also that it cannot be interacted with by the user. With relation to requirement 5, the persona graphic has an animation for its mouth to denote that it is the source of all audible communication.

8. The system's mode of communication with the user should have a 'permanent' nature.
Design Solution: Key words to summarise the most salient aspects of the instruction or feedback are printed to the 'key-word-box' below the interface, providing a reference for the user. Being located below the interface persona the relationship between the audible communication and the key-word box is reinforced. This also means that a particular location in the interface is defined for delivering all information, both audible and textual to the user.
We can also recognise that the repeat button (requirement 6) complies with this requirement as it allows the user to repeat any audible instruction or feedback on demand and thus at any time.

9. The system should provide appropriate feedback to the user.
Design Solution: When the user enters a wrong answer to a question in the task, quantitative feedback is given to give some direction on where they went wrong. The user can then apply this in a 'second chance' to answer the question. If they subsequently get this wrong, further feedback is given by revealing the correct answer, so to consolidate the child's learning.
By involving the foundation stage teachers in the design process it was possible to accurately reflect the language used in the classroom for the purposes of giving appropriate feedback. This also ensured that it is familiar to the children, used in the correct mathematical context and closely related to the national curriculum.

10. The system should allow the user to control all tasks.
Design Solution: The user can interrupt the interface persona's audible instructions when it is delivering the task instructions or feedback. This means that they are not 'locked out' of the interface and are able to control the pace at which the task is performed. This design solution also relates to requirement 11.

11. The system should allow the user freedom to self direct the learning session
Design Solution: When the child completes the task they can choose whether to repeat the task, increment the level of difficulty or end the task thus enabling them to self-direct the learning session in accordance with their desires. Also they have the freedom to quit the task at any time using the 'return to main menu' button, reducing the propensity of the system to 'trap' the user.

12. The system should have a reward scheme, providing a goal for the learning session.
Design Solution: The user is awarded a gold coin for each question answered correctly. These are accumulated at the top of the screen in the progress indictor, which allows the user to assess how they have performed throughout the task. The progress indicator also provides some visibility of system status, such that the user can recognise 'where they are' in relation to finishing the task.
At the end of the task the total number of gold coins collected relates to a piggy bank animation, which denotes the child's achievement (i.e. the more coins accumulated the

more elaborate the animation) and provides a sense of fun. It also provides a sense of closure as the user can recognise that the task is finished and can prepare for the next task (relating to usability principle, Dix et al, 2000).

13. The system should provide a printable summary of the questions answered.
Design Solution: At the end of the task the user can print a summary of their lesson, which details the questions posed throughout the task and whether the user gave the correct answers.

14. The system should provide a set of teacher controls to allow appropriate customisation of the system interface and the tutorial content.
Design Solution: Customisability options are available in a teacher controls screen, which is launched from a button on the main menu (relating to requirement 1). This button is specifically for the teacher and not for child users, hence its placement away from task related buttons, in the main menu.
In the teacher controls screen the teacher can customise the appearance of the main menu to either display:
    - A single button to the task, where the three levels must be performed temporally.
    - Three buttons, one for each level of the task, so that the user can perform them in any order, at any time.

    This allows the teacher to toggle the system between a temporal task structure (where levels have to be completed in sequence) and an options structure (where levels can be completed at will), so to suit the activities of the classroom.
The exit from the system is also in the teacher controls area, so that it is under the control of the teacher.

3.2 Non-functional requirements

15. The system should be an effective teaching tool allowing children to practice applying mathematical concepts taught in the foundation stage classroom.
Design Solution: The 3 levels of the system's task are closely related to lessons that are given in the classroom and thus clearly relate-to and fulfil important objectives of the national curriculum. The levels incrementally increase in difficulty thus providing the opportunity for the user to expand their knowledge in more difficult contexts. The user can also repeat any of the levels without the same data being reused.

16. The lesson length should be approximately 20 minutes.
Design Solution: With relation to requirement 15, the task provides 3 levels and the user can repeat any of these without the same questions being posed in the same order. Consequently the learning session can comfortably span the required 20 minute period and is clearly very flexible.

17. The system should be reliable and not produce any unexpected behaviour.
Design Solution: With relation to requirements 23 and 30, the interface has been designed to 'fill' the screen so to prevent users from launching actions outside the scope of the system. The interface also uses a consistent naming convention for all buttons as well as icons where appropriate, so to promote recognition of the function they represent and thus minimise navigational errors. Moreover buttons with comparable navigational actions have been placed consistently on all screens of the interface so to promote predictability and prevent user surprise when they are launched.
The system should be rigorously tested before deployment in the classroom to ensure that this requirement is upheld.

18. The system must be able to run on the on the classroom computers.
    Design Solution: The size of the interface has been designed to be compliant with screen resolution of classroom computers at 800x600 pixels. This requirement should be further fulfilled as part of the system's implementation, such that thoughtful decisions are made to comply with the system's hardware constraints.

19. The system should be capable of running as a standalone program.
    Design Solution: This requirement should be fulfilled as part of the system's implementation.

3.3 Usability and user interface requirements

20. The system should instantiate a graphical direct manipulation interaction style.
    Design Solution: All user input is performed using point and click operations, including the navigation of the interface and the entry of the user's answer to questions in the task (as this is multi-choice). This relates to user requirement 29 such that the mouse should be used as the key input device, in consideration of children's computing skills.

21. The interface should allow direct manipulation of objects for relevant tasks, so to provide a different perspective on how to arrive at an answer in a mathematics context.
    Design Solution: This requirement has not been upheld as it was decided that point and click input operations should be used to allow the user to select their answer to questions, thus supporting the multi-choice format. Moreover the graphics that form the learning metaphor are meant to be used as an application for calculating the answer and accordingly it was decided that these should not be manipulated by the user.

22. The user interface should be easy to understand, operate and navigate.
    Design Solution: Iconic buttons have been used where appropriate in the interface to promote recognition of the function they represent. Also, navigational buttons that change the scope of the interface have been consistently placed on each screen so that the user should be able to predict the navigational action launched by clicking certain buttons. Moreover all buttons are appropriately sized and have 'effects' when they are rolled-over, hit and down so to provide feedback to the user on whether they are on the button hotspot, or have performed an action upon a button.
    The task is also graphics intensive so that it is appealing and easily interpretable for the user.

23. The user should not be able to click any buttons or hyperlinks outside the scope of the tutorial area.
    Design Solution: The interface has been designed to fill the computer screen, and thus it must run at 800x600 pixels, so to comply with the hardware constraints of computers used in the foundation stage classroom.

24. The interface should be highly learnable, allowing users to easily pick up how to use the system.
    Design Solution: The interface has a set of standard features that are applied to all screens to ensure that the design principles of consistency and familiarity are upheld (Dix et al, 2000).
    In relation to requirement 20, all user input is performed using point and click operations meaning that the user can apply their knowledge of interacting with the system in all areas of the interface.

25. The system should allow a child to work alone, without the need for teacher guidance.
    Design Solution: Audible communication has been used so that instructions and feedback can be easily interpreted by the user. This can be repeated on demand if the child does not understand it on first listening. A text box that denotes the key points of the instructions and feedback also provides a reference on the key points of the instruction for the user, meaning that they don't have to disturb the teacher if they are confused.

26. The system should be accessible to more than one user at a time.
    Design Solution: Instructions and feedback that are delivered through the interface are not user specific and thus they can be interpreted by more than one child so to support learning in collaboration. Moreover the tasks, by their nature, are accessible to more than one child user such that children can discuss each question and decide on their answers.

27. The system should be able to be used in conjunction with the other computer-based tutors used in the classroom.
    Design Solution: The task has been designed thoughtful so that the mathematical concepts that are presented and the language used are closely mapped to the national curriculum. This was aspect of the system design was directed by the foundation stage teachers to verify that it is appropriate. Accordingly it should not conflict with the lessons of other systems.

3.4 User requirements

These requirements capture the characteristics, abilities and skills of the intended user group:

28. The system should support the user's level of mathematics competency.
    Design Solution: With relation to requirement 27, the task was iteratively designed with the help of foundation stage teachers to ensure that the mathematical content of the task was appropriate for the foundation stage. Also the levels of the task directly reflect the 'number in the bag' and 'introduction to addition' lessons that are used in the classroom, and accordingly they are closely mapped to the national curriculum.

29. The mouse should be used as the key input device for interacting with the computer.
    Design Solution: With relation to requirement 20, point and click operations are used to perform all operations in the interface as well as user input, and thus the mouse is used as the only input device for interacting with the system.

30. The interface should allow a large margin for error to compensate for the user's potentially limited control over the mouse and thus the screen pointer.
    Design Solution: All buttons in the interface are suitably large in size and are well spaced on each screen of the interface to prevent potential user errors given their potential limited control over the mouse (this relates to requirement 17). Also roll-over effects have been specified as necessary for buttons so that the user is aware that they are 'on' a button so to prevent confusion.

31. The system should not require the user to learn any new skills, and thus it should be tailored to their skill set and computing experience.
    Design Solution: Issues of multitasking are avoided in the system, as each question of each level is performed temporally. Also from the requirements analysis chapter we noted that all the children have experience of performing point and click operations at the interface through interacting with other computer systems in the classroom. Therefore they will not have to learn any new modes of interaction.

Implementation Appendix

1. Flash MX and ActionScript
2. Flash Development Environment
3. Animations
4. Navigational Structure
5. Functions
6. Testing

1 Flash MX and ActionScript

Macromedia Flash is a multimedia graphics program that supports the development of interactive 'movies'. Flash programs are called 'movies' as they share many attributes with standard movie playback. They are constructed with frames that depict moments in time, and are played by a play-head at a specified frame-rate per second. Therefore when played from a start-point to an end-point it is possible to see that moving a graphic between two points or fading a colour between two shades is performed by setting the start and end points and then making incremental changes in the frames between these points to effect the animation.
The ActionScript programming language can be used to add code to any frame in the movie to specify 'stopping' points, loops and conditions in the playback and thus support user input and interactivity.

Flash has many tools to aid the development of graphics and artwork, as well as animation for which it has a number of built in functions that can automatically set the intermediate frames for an animations. It is also possible to attach sounds to particular frames in animations or interface screens if required.

Flash uses layers to build up an image or scene[30]. This allows the author to define a background layer, a sound layer, a buttons layer, a coding layer and so on, to separate salient aspects of a scene, animation or graphic and specify the order in which they appear. This provides the flexibility to work on particular layers without modifying others and also to move layers on top of others, which is important in very compiling complicated scenes, animations or graphics.

Flash uses a component library to store graphics, animation and sound components, meaning that components are easily accessible for editing and reuse. This also means that multiple, identical instances of a graphic or animation can be used in the interface, and can all be modified from one location.
Components of the flash library can also be imported into other flash files. This is important for the purposes of future extension as clearly it would be necessary to extend the high fidelity prototype implementation of the system if it was to be prepared for release with many tasks for the user to perform.

A further advantage of Flash is that it uses vector graphics, which means that all scenes, graphics and animations are resolution-independent and therefore it is possible to rescale them to any size without losing clarity or quality. Also vector graphics are more lightweight than comparable quality bitmap images. This means that all flash programs are highly portable and can be run on displays with a variety of screen resolutions (not just the designed 800x600 pixel resolution). Moreover, a 'full-screen' parameter can be set on flash movies to ensure that they fill the screen when run within the flash player. This complies with requirement 23 as clearly when at full screen projection the user cannot click on any buttons or hyperlinks outside the scope of the tutorial area.

As previously stated ActionScript is flash's own programming language, which is similar in many ways to JavaScript. It can be attached to any frame in a scene or animation to support more advanced functionality and interactivity. ActionScript is object-oriented and allows the author to define functions which are similar to methods in Java terms. These can be used to store data, execute particular actions or set conditions on the movie playback for example. This enables the development of sophisticated software products that support user input, data storage and interactivity.

[30] Flash terminology for a screen of the interface.

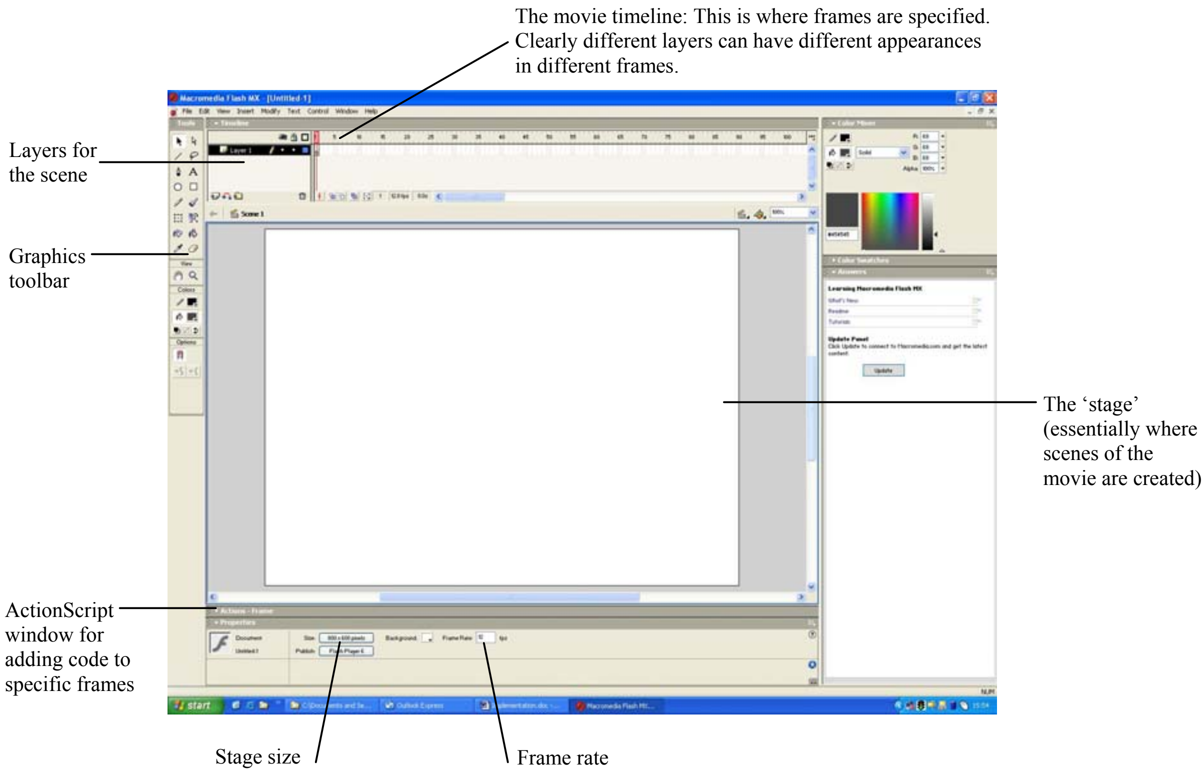
Flash is usually associated with web-based delivery and thus can be easily embedded within an HTML webpage and using the flash browser plug-in. This is not necessary suitable for the high-fidelity prototype, given the need for the user to not have to download any extra software to run the system (non-functional requirement 19).
However, flash movies can also be published in '.exe' format meaning they can be deployed and run from an executable file on any Windows or Macintosh compatible computer system[31]. Using this method of publication flash is packaged with its own flash player, meaning that it is projected correctly and predictability every time it is run. Although this does not provide platform independent flexibility, this clearly complies with the hardware constraints specified in non-functional requirement 18.

Macromedia Director with ActionScript is a more complete package for building applications with all the capabilities of Flash for embedding animation, graphics and sound. Director allows an application to be developed, published and deployed anywhere, as a standalone program that is independent from a media player for playback (similar to a Java application). It also supports the production of installation scripts, meaning that applications can be installed from a CD and then run at any time.
Director was used to develop the Tizzy's Toybox and Teddy Bear's Picnic systems which were evaluated in the requirements analysis chapter. However, it is inaccessible for the purposes of this project due to cost and availability constraints. As already stated though, Flash provides exactly the same support for developing an interactive movie, with differences in the methods for publication only, and thus it is well suited to developing the prototype solution.

[31] This complies with non-functional requirement 19, which states that the system should be capable of running as a standalone program.

2 Flash Development Environment

Figure 1: Screen shot of the flash development environment:



3 Animations

Figure 1: Frog animation (jump on spot)

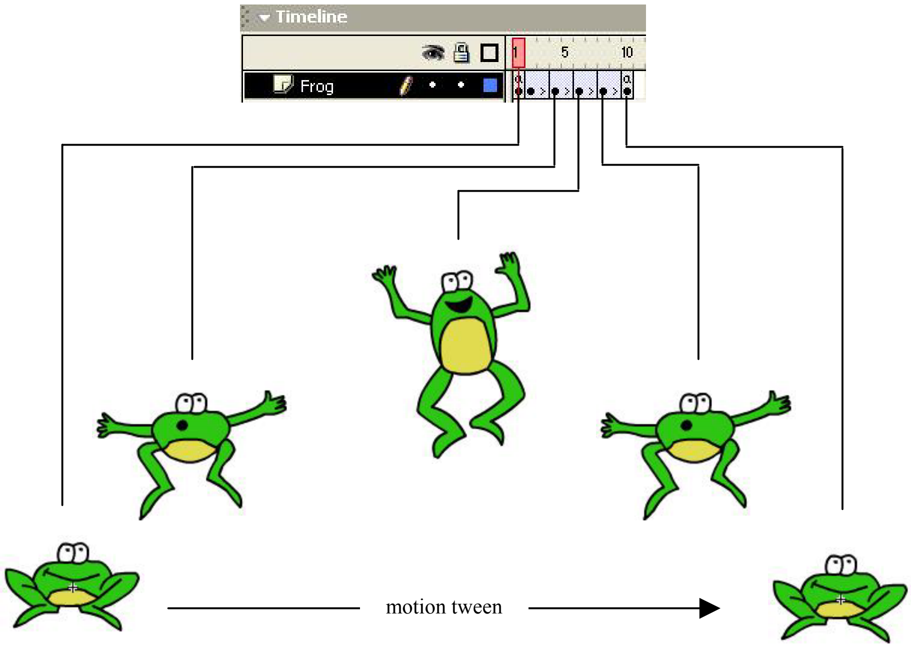


Figure 2: Interface persona animation

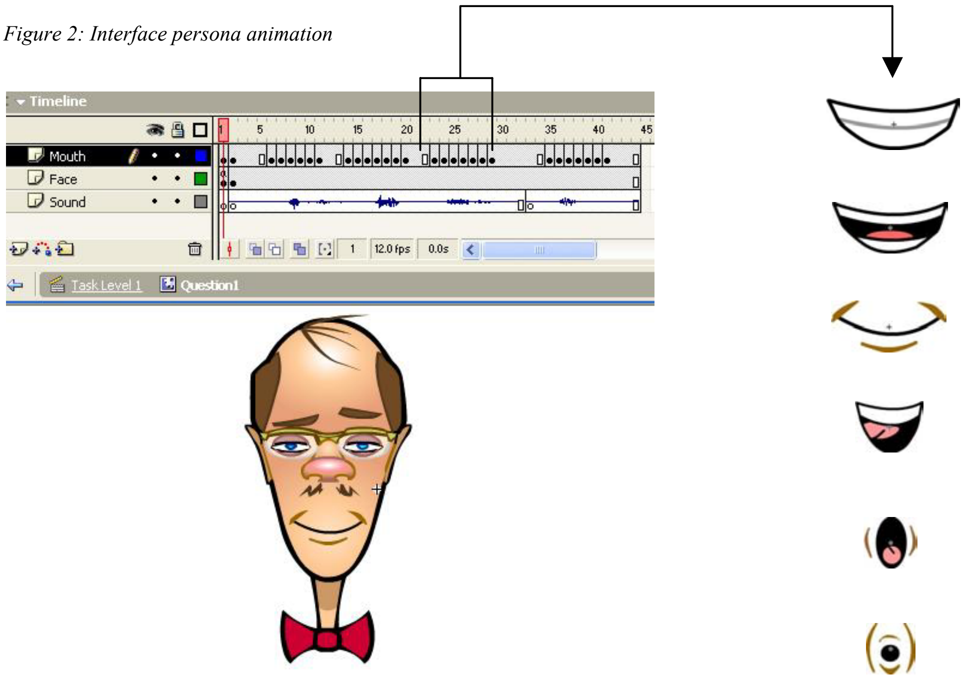
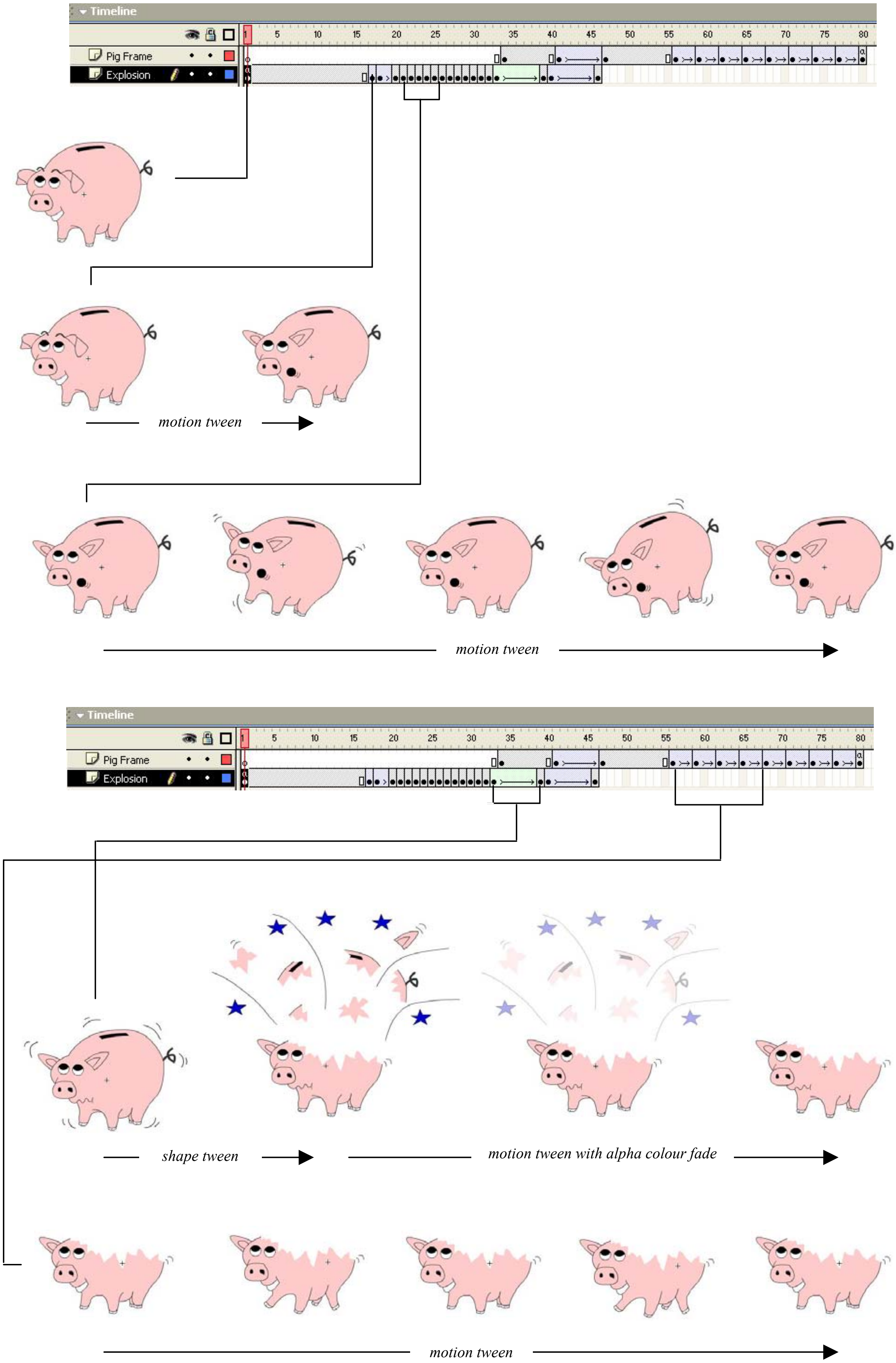


Figure 3: Piggy Bank animation ('exploding')



4 Scene Definitions

1 Main Menu

The layers that are associated with the main menu are given below. Clearly there are small differences in the content of these layers for the temporal and options structure, but for the purposes of this scene definition it is not necessary to explain these in detail here.

- Dynamic text: This layer contains a dynamic text field called `output`. This is used for text output of key words to screen.
- Navigation: This layer contains the navigational button instances relevant to the main menu:
  - Button to teacher controls: On release this button loads the `tutor_teacher_controls.swf` to level 0, replacing the current main menu movie.
  - Button to the 'frog in the pond' task level: On release this button loads `tutor_level_1.swf` to level 0, replacing the current main menu movie.
- Repeat Button: This layer contains an instance of the repeat button for the audible output. On release this button plays the interface persona movie using the `eval.(Persona).gotoAndPlay()` function.
- Persona: This layer contains an instance of the `choose_a_game` persona movie clip and has the attached sound `choose_a_game.wav`. It also contains the movie clip placeholder for the interface persona which is used as a reference point to ensure that the movie is placed consistently every time it is played.
- Code: This layer contains the frames `Init_MM` and `Standard_MM` and their associated Action Script coding.
- Scene (folder): A layers folder contains a number of related layers. In this instance it contains three sub-layers that make up the scene:
  - House: This layer contains the house graphic
  - Interface Frame: This layer contains an instance of `Interface_frame`.
  - Background: This layer contains a rectangle that fills the stage area. It is filled with a gradient that blends white with a light shade of green. The white gradient starts behind the interface persona to indirectly denote its importance in the scene

2 Teacher Controls

2.1 Teacher Controls Scene

The layers that are associated with the teacher controls scene are given below. Clearly there are small differences in the content of these layers for the temporal and options structure, but for the purposes of this scene definition it is not necessary to explain these in detail here.

- Controls: This layer contains:
  - Button to teacher controls: On release this button loads the `tutor_teacher_controls_options.swf` to level 0, replacing the current teacher controls movie.
  - Static text boxes to denote the two available options for the user to switch between; the temporal structure or the options structure.
  - A 'tick' graphic. To denote that the temporal structure is currently selected.
- Code: This layer has a frame that enacts the `stop()` function, and thus stops the movie play-head when opened.
- Scene (folder): This folder contains three sub-layers:

  - Navigation: This layer contains:
    - Button to exit the system: On release this button moves to, and stops the movie playback at the `exit_confirmation` scene, which is a scene contained in this movie.
    - Button home: On release this button loads the `tutor_MAIN.swf` to level 0, replacing the current teacher controls movie.
  - Interface frame: This layer contains an instance of `Interface_frame_teacher_controls`.
  - Background: This layer contains a rectangle that fills the stage area. It is filled with a gradient that blends white with a light shade of purple. The white gradient starts in the middle of the scene and deepens towards the boarder of the interface frame.

2.2 Exit Confirmation Scene

This scene is identical for both the temporal and task structure:

- Exit Door: This layer contains a graphic of an open exit door (to promote user recognition that the system will close if they click 'yes' to close the system).
- Scene (folder): This folder contains three sub-layers:
  - Navigation: This layer contains:
    - 'Yes' button to exit the system: On release this button enacts the flash `fscommand` to close the current movie and the flash player.
    - 'No' button to return to the teacher controls: On release this button enacts the `gotoAndStop()` function to move to and stop the movie play-head at the teacher controls scene.
  - Interface frame: This layer contains an instance of `Interface_frame_teacher_controls`.
  - Background: This layer contains a rectangle that fills the stage area, which is filled with the same fill-effect as the teacher controls scene to promote familiarity (Dix et al, 2000).

3 Task

The layers that are associated with each level of the task are given below. Clearly there are small differences in the content of these layers for levels 1, 2 and 3, and for the temporal and options structure, but for the purposes of this scene definition it is not necessary to explain these in detail here.

- Animation Mask: This layer is a mask for the animations. A mask can be used to hide part of the stage or its surrounding area and thus has been used to ensure when the animations are not *on the stage* - they are not visible. The layers beneath the mask are:
  - Piggy Bank: This layer contains the piggy bank animations and the related placeholder. These movies are used in the reward scheme.
  - Gold Coins: This layer contains the gold coin animations and the related placeholders. These movies are used throughout the task and in the reward scheme.
- Dynamic text: This layer contains a dynamic text field called `output`. This is used for text output of key words to screen.
- Questions: This layer contains the question text for each question of the task and the related placeholder. The question text is used for each question posed in the task and is positioned at the bottom of the screen *only* when the related question is played
- Frogs: This layer contains the frog animations and the related placeholders. These movies are used throughout the task.

- Repeat Button: This layer contains an instance of the repeat button for the audible output. On release this button plays the interface persona movie using the `eval.(Persona).gotoAndPlay()` function.
- Task Buttons: This layer contains an instance of each of the lily pad buttons (1 to10) that are used for user input to the questions. On release each of these button instantiates the `ButtonCondition()` function (refer to section 4) which assesses whether the user has input the correct answer.
- Task Graphics: This layer contains the graphics that make up the frog in the pond learning metaphor, including the pond, reads, water, and so on.
- Persona: This layer contains all instances of the persona movie clips relevant to the level of the task. It also contains the movie clip placeholder for the interface persona.
- Functions: This layer contains the user defined functions that are called in the appropriate frames of the code layer. Refer to section 4 for an explanation of each of these functions.
- Code: This layer contains the frame labels for all frames of the movie and the associated Action Script coding.
- Scene (folder): This folder contains three sub-layers:
  - Navigation: This layer contains:
    - Button home: On release this button loads the `tutor_MAIN.swf` to level 0, replacing the current task movie.
  - Interface frame: This layer contains an instance of `Interface_frame_task`.
  - Background: This layer contains a rectangle that fills the stage area. It is filled with a gradient that blends white with a light shade of orange, consistent with the teacher controls scene.

4 Functions

The functions layer for each level of the task contains the object oriented methods that are called in the associated frames. A description of each function that was written is given below:

Function	Description
Sleep(time)	Stops the movie playback before restarting it after 'time'.
InitQuestionArray()	Initializes three arrays: 1) holds the question indexes for questions 1-8 2) holds the answers to the questions 3) holds the relevant keywords for the questions
InitPlaceholders()	Initialise the arrays for holding the placeholder coordinates. Placeholders are used to place individual movie clips throughout the task level, including persona movies, frog movies, and the gold coins for the reward scheme.
QuestionOrder()	Defines a GameQuestions array that holds the order of the questions for the level, and randomises them to ensure that they are not in sequential order.
SetUpQuestion(CurrentQuestion)	Sets up the specific parts of the question for the appropriate question id. This includes setting up the persona movie, the frog movie, the question text, the key word text, the progress indicator and also writing the question to the summary sheet.
SetUpAnswer(CurrentQuestion)	Sets up the specific parts of the answer for the appropriate question id. This includes setting up the persona movie, the frog movie, and setting the key word text.
AnswerReturn(CurrentQuestion)	Returns the answer to the current question.
CleanUpAll()	Removes all individual movie clips from the current scene and sets the key word text to nothing.
ButtonCondition(name, answer)	Button condition for user input. This is called when the user clicks one of the task buttons when answering a question. It writes the user's input to the summary sheet, and then evaluates

	whether it is right or wrong. If wrong initially the user is allowed to try again. (Arguments: name = button name, answer = question answer)
TooFew()	Plays the persona movie to denote that the answer input is too few and sets the key word text accordingly.
TooMany()	Plays the persona movie to denote that the answer input is too many and sets the key word text accordingly.
CorrectAnswer(CurrentQuestion)	Called when a correct answer is given to the current question. Sets the key word text, award a gold coin, plays one of three persona movie to denote "correct", writes the result for the question to the summary sheet.
IncorrectAnswer(CurrentQuestion)	Called when an incorrect answer is given to the current question. Sets the key word text, plays one of two persona movie to denote "wrong", writes the result for the question to the summary sheet.
InitPlaceHoldersReward()	Initialises the arrays for holding the placeholder coordinates for the reward scheme, including the piggy bank animation.
EndOfLevel(NumberCorrect)	Sets up the end of level scene. Writes the number correct (out of 8) to the summary sheet, sets the key word text to a response that befits the child achievements, sets the dynamic text box to the number of questions correct.
SetUpRewardScheme(NumberCorrect)	Sets up the specific parts of the reward scheme depending on the number of questions answered correctly. This includes setting up the persona movie given the placeholder coordinates, and piggy bank animation that relates to the child's achievements.
SetUpNextStage()	Sets up the specific parts of the next stage scene. This includes setting the key word text and the persona movie.

5 Navigational Structure

The network diagrams for the temporal structure and the options structure are shown in the following figures:

- Figure 1: Temporal structure overview - including level 1
- Figure 2: Temporal structure – level 2
- Figure 3: Temporal structure – level 3
- Figure 4: Options Structure – all levels

Figure 1: Navigational structure (temporal)

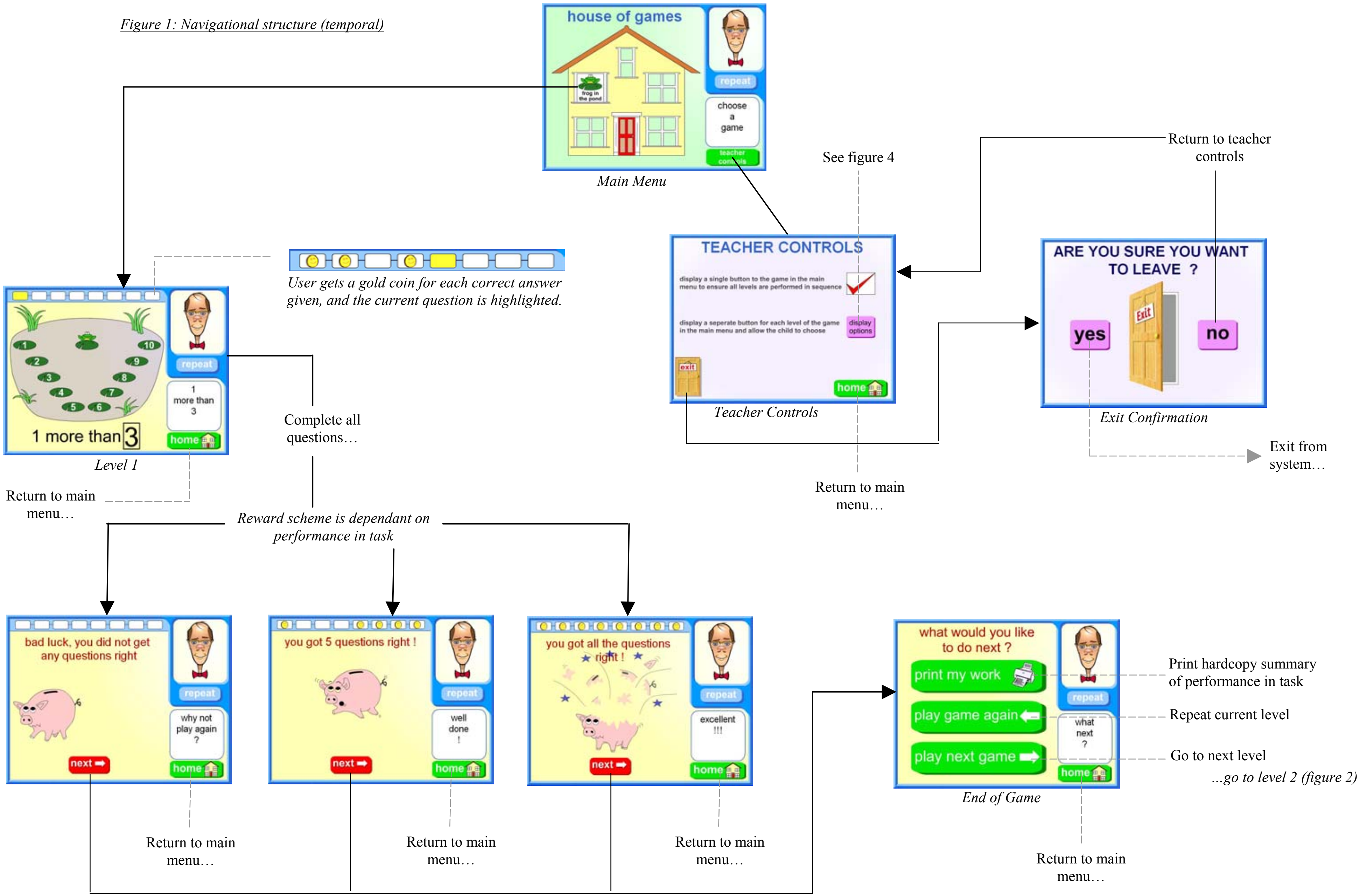


Figure 2: Navigational structure level 2 (temporal)

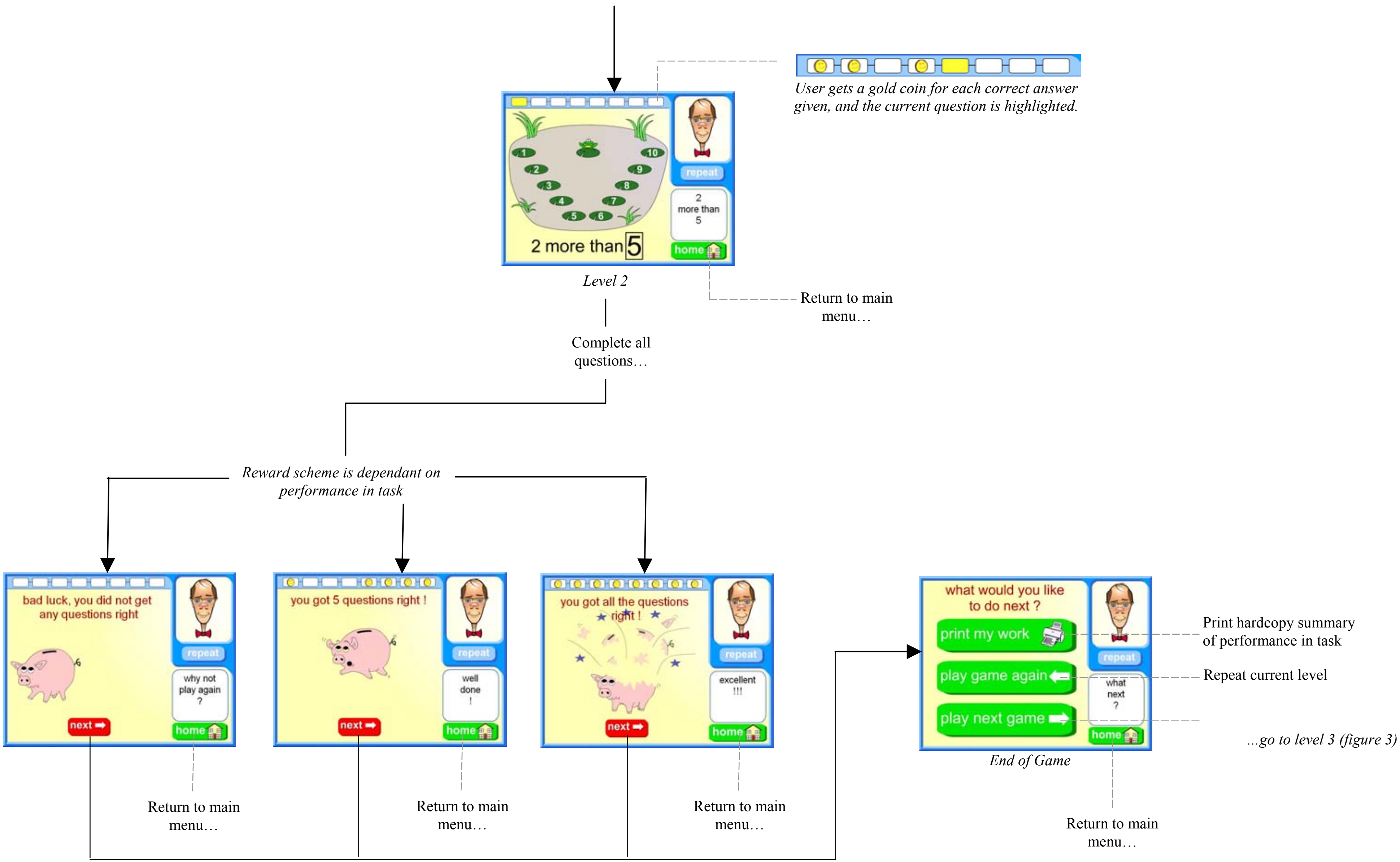


Figure 3: Navigational structure level 4 (temporal)

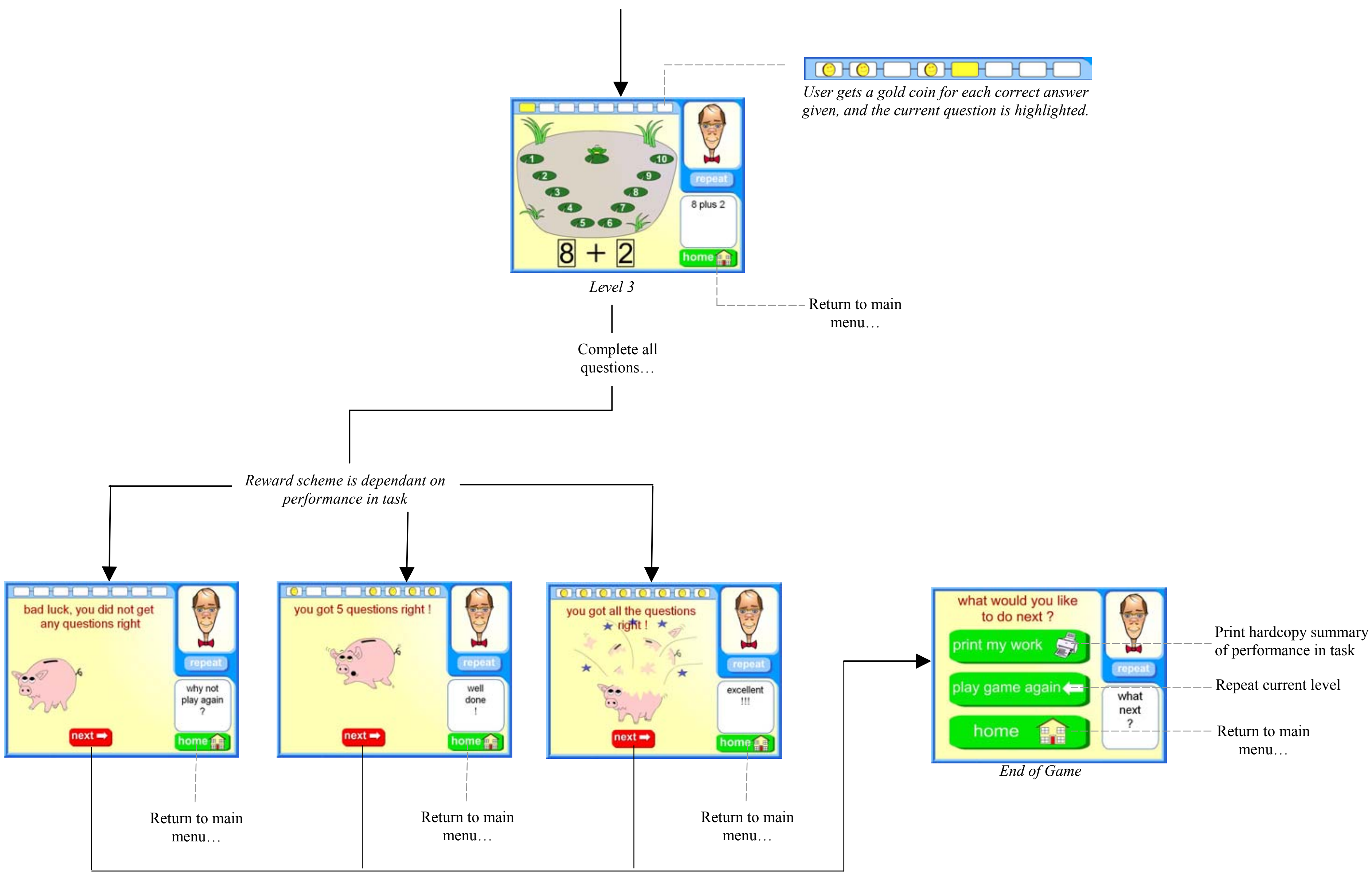
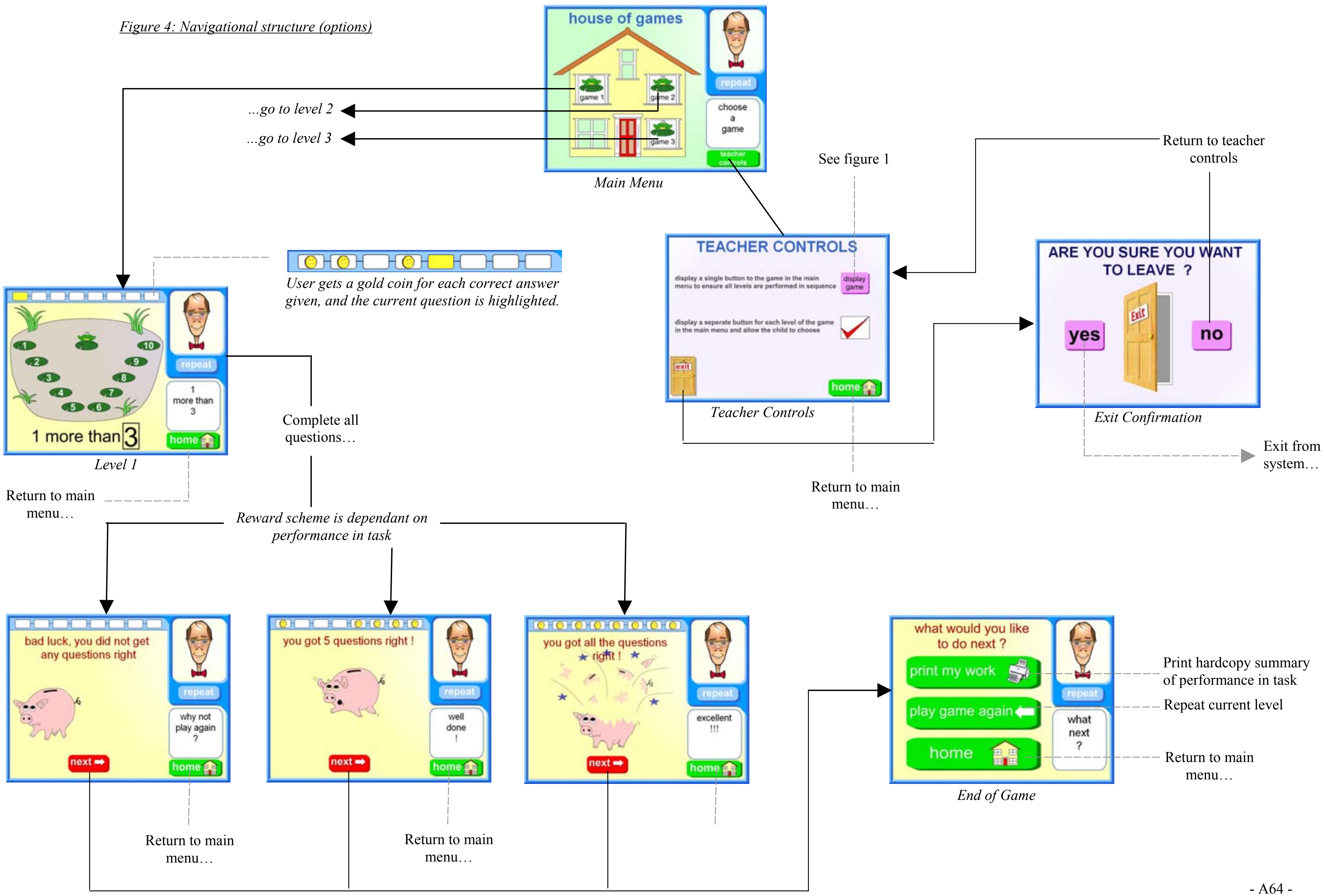


Figure 4: Navigational structure (options)



6 Testing

This section denotes a sample of the tests performed on the system, using black box techniques. The tests shown were performed on task level 1.

Test ID	Test	Expected Result	Actual Result
9	Test whether the questions for level 1 are randomised by opening the task on 3 separate occasions from the main menu.	The questions should be posed in different order each time.	Expected result upheld. Question order observed: 1) 2,5,6,3,1,7,4,8 2) 7,3,2,5,6,8,1,4 3) 1,4,3,8,2,6,7,5
10	Give a correct answer to a question	The persona should give one of the responses for a correct answer. The correct answer should then be demonstrated. The user should be awarded a gold coin.	Expected result upheld
11	Give an answer that is too few, in anticipation of a try again.	The persona should give the feedback “too few”. The key word box should also denote this. User should be able to try again.	Expected result upheld
12a	Give a correct answer using the try again	The persona should give one of the responses for a correct answer. The correct answer should then be demonstrated. The user should be awarded a gold coin.	Expected result upheld
12b	Give a wrong answer using the try again	The persona should give one of the responses for a wrong answer. The correct answer should then be demonstrated.	Expected result upheld
13	Give an answer that is too many, in anticipation of a try again.	The persona should give the feedback “too many”. The key word box should also denote this. User should be able to try again.	Expected result upheld
14a	Give a correct answer using the try again	The persona should give one of the responses for a correct answer. The correct answer should then be demonstrated. The user should be awarded a gold coin.	Expected result upheld
14b	Give a wrong answer using the try again	The persona should give one of the responses for a wrong answer. The correct answer should then be demonstrated.	Expected result upheld
15	Click the button to move to the next question	The next question in the randomised sequence should be posed.	Test failed. Playback moved forward too quickly and caused an error.
15 (test repeated after changes)	Click the button to move to the next question	The next question in the randomised sequence should be posed.	Expected result upheld. Given the results from test 9 we observed the following question order: 1) 2 followed by 5 followed by 6, etc. 2) 7 followed by 3 followed by 2, etc 3) 1 followed by 4 followed by 3, etc

Evaluation Appendix

1. Results Summary
2. Graphical Representation of Results

1 Results Summary

Participant 1

	Tuesday		Friday	
Level	**Right**	**Try Again**	**Right**	**Try Again**
1	8	0	8	0
2	6	7	7	1
3	6	4	6	2

Participant 2

	Tuesday		Friday	
Level	**Right**	**Try Again**	**Right**	**Try Again**
1	8	1	8	1
2	8	3	8	1
3	8	1	8	2

Participant 3

	Tuesday		Friday	
Level	**Right**	**Try Again**	**Right**	**Try Again**
1	6	4	8	0
2	3	7	6	7
3	4	7	5	5

Participant 4

	Tuesday		Friday	
Level	**Right**	**Try Again**	**Right**	**Try Again**
1	8	2	8	0
2	7	6	7	2
3	6	2	7	1

Participant 5

	Tuesday		Friday	
Level	**Right**	**Try Again**	**Right**	**Try Again**
1	8	0	8	0
2	7	6	8	3
3	6	3	8	0

Participant 6

	Tuesday		Friday	
Level	**Right**	**Try Again**	**Right**	**Try Again**
1	8	3	8	2
2	4	4	6	3
3	5	5	7	2

2 Graphical Representation of Results

Figure 1: Level 1: Comparison between Monday and Friday

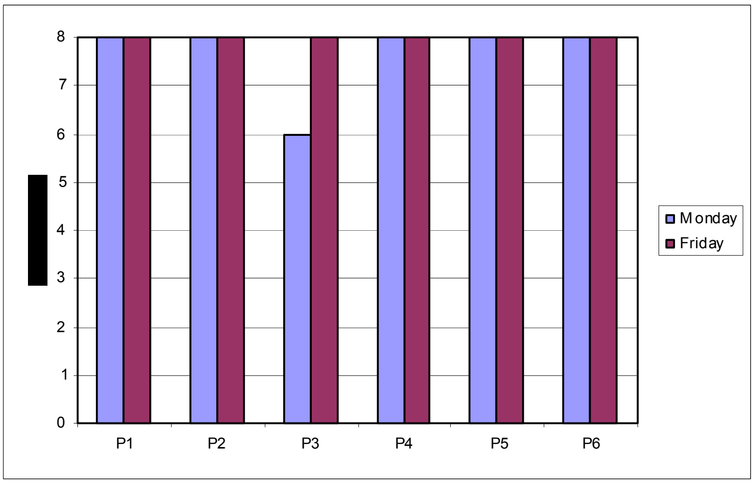


Figure 2: Level 2: Comparison between Monday and Friday

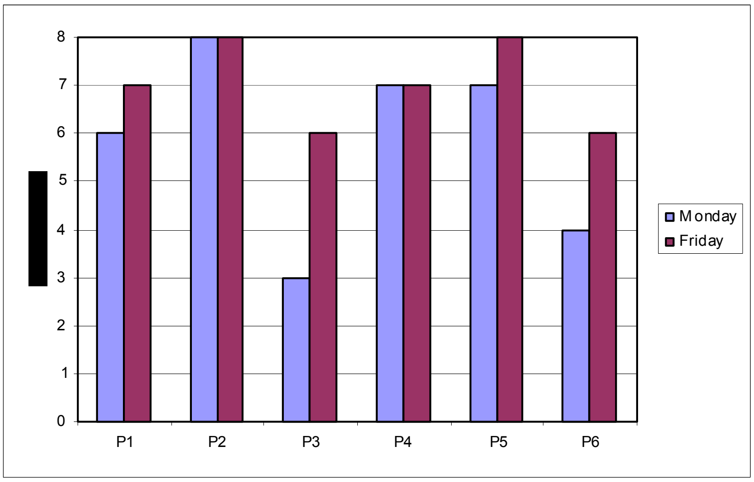


Figure 3: Level 3: Comparison between Monday and Friday