The InitiaL Rider Training Project

An evaluation of the potential of

e-Coaching for Riders

Version 1.01

Tampere University of Technology
Hypermedia Laboratory

Pekka Ranta
Antti Mäki
Miika Huikkola

April 2007

A project from the Federation of European Motorcyclists' Associations

FEMA
Federation of European Motorcyclists Associations

TABLE OF CONTENTS

FOREWORD

We would like to thank Robert Tomlins, FEMA Assistant General Secretary and IRT Project Co-ordinator for his invaluable ideas and vision of how e-Coaching could be applied to initial rider training. We would also like to thank Commander Roger Renoy, head of the Belgian Police Driver and Rider Training Academy and President of the Belgian Motorcycle Council and Marc O'Loideoin, Senior Instructor of the Star Rider Training Scheme of Ireland for their comments and participation. Our work on this evaluation relied heavily on their expertise in the domain of motorcycle rider training.

Pekka Ranta, Antti Mäki and Miika Huikkola,

Tampere University of Technology,

Hypermedia Laboratory.

Tampere, April 19th, 2007.

1. INTRODUCTION

1.1 Project Purpose

ACEM, FEMA and FIM agree that initial rider training arrangements in Europe often fail to meet the needs of the rider. One objective of the IRT Project is to evaluate the contribution that e-Learning, herein after referred to as e-Coaching, could make in developing hazard perception and avoidance and correct attitude and behaviour in initial rider training. This objective would be met through an affordable structured programme, created by competent instructors that would expose trainees in a no risk context to hazards which they are likely to meet in traffic.

An accessible e-Coaching programme, available at no or little cost, which can be run on readily available personal computers or video game consoles, would be an unrivaled solution when trying to reach the two million initial riders throughout the European Union. Since the motorcycle accidents have such high social and personal costs, reducing the incidence of these accidents would be worth the money and effort invested.

The goal of this document is to give a technical evaluation of the feasibility of such an approach and to make recommendations on how to develop and implement such e-Coaching software and provide an initial estimate of the costs of development and implementation.

1.2 Project Associates

This feasibility study is a joint undertaking within the Initial Rider Training (IRT) Project between the project and the Hypermedia Laboratory of Tampere University of Technology (TUT). The views and expertise in the domain of motorcycle riding offered by the members of the IRT Project has been an invaluable aid in identifying the case-specific features of an e-Coaching approach. Based on these views and the study of the existing scientific research and similar projects the Hypermedia Laboratory has produced this evaluation.

1.3 Document Purpose

This document is written to address the questions raised by e-Coaching as a project. The document considers the technical aspects of the e-Coaching approach, such as the required features of the system and the most suitable implementation technologies for these features. The document is incorporated to the supervisory board's report to EU Commission.

1.4 Document Overview

Chapter 2 describes why an e-Coaching evaluation was launched, the potentials of such an approach and what the goals of this evaluation are.

Chapter 3 gives a short introduction to the e-Learning environments and simulations and discusses the benefits they offer for initial rider training.

Chapter 4 defines the work methods used through this evaluation.

Chapter 5 lists the identified requirements for an e-Coaching programme.

Chapter 6 discusses the suitability of PCs and video game consoles as platforms for e-Coaching software. The advantages of Internet-based approach are reviewed.

Chapter 7 contains the synopsis for an IRT e-Coaching programme and goes through the requirements of the user interface. Chapter also describes the architectural model and process description of the system. The topic includes what parts the programme consists of and how do these parts interoperate. It is also suggested how the introduced components could be implemented.

Chapters 8 offers rough cost estimates for the different implementations of the IRT e-Coaching programme.

Chapter 9 contains the foreseen potential challenges and opportunities.

Chapter 10 summarizes the conclusions reached in the document.

Appendix A clarifies the definitions of the technical terms, acronyms and abbreviations, while appendix B lists the publications referenced throughout this document. Appendices C and D offer illustrations of the architectural design and process description, respectively. Appendix E describes some of the scenarios which were used to define the requirements for the IRT e-Coaching programme.

2. OBJECTIVES OF THE PROJECT

The quality of motorcycle rider training varies greatly within the European Union. The availability of trainer services, competency of the trainers and the cost of the training varies from country to country. Even in the countries where good, affordable training is available, the training tends to concentrate on machine controls, often not adequately addressing matters like hazard perception and avoidance as well as correct attitude and behaviour in traffic.

As a part of the IRT Project launched by ACEM, FEMA and FIM, supported by the European Commission, Vägverket the swedish national road traffic authority and IVV the international driving instructors´ organisation, a computer based simulation is being considered as one of the methods to overcome these shortcomings in many current initial rider training arrangements. Such an e-Coaching approach will need to be an affordable solution working on the users' existing PCs or video game consoles. The e-Coaching approach concentrates on the aforementioned shortcomings in credible virtual traffic situations thereby exposing the rider to hazards without risk and leaves the on-motorcycle training to the real-life instructors.

2.1 Target Group

Two groups can be identified among the potential users of a rider e-Coaching programme, the gamers and the non-gamers.

The gamers are young, approximately 15-35 years old people, who are accustomed to video games through a lifelong experience, but have little or no experience in real-life traffic. They are likely to have no difficulties learning the controls of an e-Coaching programme. In addition to the educational aspects, they are also likely to an e-Coaching programme like a video game and the challenges it poses.

The non-gamers are older, perhaps middle-aged born-again bikers. They have less experience on video games, and are likely to use an e-Coaching programme just see what is has to offer as an aid to safer riding, not to spend time with it like a video game. Unlike the gamers, the non-gamers are likely to have previous experience in traffic, maybe even with powered two wheelers.

To satisfy the needs of the both user groups, an e-Coaching programme would be designed to have a level-based approach. The first levels are quite easy, thus allowing the non-gamers to practice their skills. As the user advantages in the levels, the exercises get more difficult and offer enough challenge to keep the gamers motivated.

2.2 Goals

An e-Coaching programme produced as a consequence of the IRT Project would need to fulfill the following goals.

- Offer a method for the trainees to experience hazardous riding scenarios, which they are likely to face in real traffic, in a no-risk context and receive advice and feedback on their performance.
- Enable learning of hazard preventing behaviour and attitude through understanding the consequences of incorrect actions, rather than reading or hearing a list of ”do nots” of motorcycle riding.
- Enable an easy transference of the learned abilities to real-life situations.
- Be a self-learning process, rather than a replacement for the traditional instructor-based training.
- Be an affordable solution even for the young initial rider trainees, i.e. work on a regular PC or video game console that the trainees are likely to have available already.
- To reach especially the young initial riders, but also the other riders who for one reason or another will not receive hazard perception training from other sources.
- Appeal to the gaming generation while remaining accessible to the older, non-gaming generation.

3. IMPROVING HAZARD PERCEPTION AND RISK ASSESSMENT WITH E-COACHING

3.1 General Psychological Factors of Rider Training

High-performance tasks are defined as complex, time-critical tasks where the operator is in the primary control loop of the system. Apart from the perceptual-motor skills, time-sharing with (subsidiary) procedural and cognitive skills for communication, orientation, and interpretation of instrument readings is required. One of the training characteristics of high-performance skills is that many people fail to develop proficiency so that selection prior to training is sometimes required. After selection, the training duration required to reach an operational level of performance usually is considerable. Typically, there are large differences between novice, advanced, and expert operators. (van Emmerik, 2004.)

According to Norwegian Public Roads Administration report (Norwegian Public Roads Administration report, 2005.) driving can be considered to be a process where the driver continuously encounters tasks in need of solutions. The ability to master these tasks will depend on the knowledge and the skills the person possesses and the person’s intellectual abilities. The objective of driver training is to remove the difference between knowledge and the skills required to drive safely, considerately and efficiently and the knowledge and the skills the person possesses when commencing training. For the training to be appropriate, the knowledge and skills that the driver needs to have must be known.

The development effort is to a large extent based on the GADGET matrix (Hatakka et. al., 1999) (the GDE matrix). This model separates what the driver must learn into the following four hierarchical levels listed from the lowest to the highest level:

1. Maneuvering level (maneuvering the vehicle)
2. Tactical level (acting in accordance with traffic conditions)
3. Strategic level (selecting journeys/trips and factors related to journeys/trips)
4. Highest level (goals for life and skills for living)

When operating with tasks at different levels, decisions made at one level can affect tasks at lower levels. When a driver at the strategic level chooses a delayed departure, he/she might wind up running late. The consequence can be that the driver at the maneuvering level must select a higher speed and utilize smaller gaps to arrive in time. This will have influence on the requirements on how to solve the tasks at the control level. This places greater demand on correct and prompt handling of the vehicle. (Norwegian Public Roads Administration report, 2005.)

It is very important to take the factors of information processing approach and cognitive psychology, especially traffic psychology, into account when designing a rider training learning environment. Traffic psychology is a young expanding field in psychology. Whereas traffic psychology is primarily related to "the study of the behaviour of road users and the psychological processes underlying that behaviour" (Rothengatter, 1997, 223) as well as to the relation between behaviour and accidents, transportation psychology, sometimes referred to as mobility psychology, has its focus on mobility issues, individual and social factors in the movement of people and goods, and travel demand management (TDM). These psychological factors are very important, because the limitations and the possibilities of our cognitive processes contribute also to our capability to use different kind of virtual tools and features. We should avoid for example cognitive load or distracting information, except when these issues are connected to educational themes and objectives.

3.2 Virtual Learning Environments

Varied virtual learning environments offer interesting possibilities to develop and enhance training and education. Learning environment means that instructional methods, resources, tools, processes, ethos and infrastructures are developed to support achieving educational goals. Virtual means that educational technologies such as web technologies, data systems, hypermedia and/or virtual reality are used. Virtual learning environments are usually implemented so that a part or whole subject matter is delivered and instructed via methods of distance learning.

It is possible to mix different methods and then we are talking about the blended learning, which is the combination of multiple approaches to learning e.g. self-paced, collaborative or inquiry-based study. Blended learning can be accomplished through the use of 'blended' virtual and physical resources. Examples include combinations of technology-based materials, face-to-face sessions and print materials. (Wikipedia, 2007d.) The main idea in intelligent computer-aided instruction (ICAI), computer-based instruction (CBI), intelligent tutoring systems (ITS), electronic performance support systems (EPSS), and adaptive hypermedia (AH) in general is to select and present the right information to the user at the right time and in the most appropriate format. So it is also possible to develop virtual learning environments where no human instruction is needed.

Technological development has been very rapid. Nowadays PC- and Internet-based technological solutions are implemented to manage demanding tasks of eLearning, which required supercomputers and dedicated networks some ten years ago. Three dimensional and real time virtual environments were used for example in military, maritime, aviation and power plant training. Now it is possible to play and train in fascinating and immersive environments at home. Interactive environments offers tools to simulate and illustrate real like and/or virtually augmented phenomena e.g. football, riding, life of communities, strategic decision making of officers and forest machines. Use of multimedia and virtual reality in PC- and console plays has been developed very rapidly and implemented in various ways.

There is no one optimal “grand theory of learning” for every context, but understanding the different points of view helps us to develop education. There are more than 50 different theories of learning [http://tip.psychology.org/theories.html]. It is also hard to try combine different frameworks into one model because educational decisions are very often compromises.

Technological possibilities help us to develop interactive and immersive learning environments, but these do not ensure efficient and high-quality learning. In order to achieve optimal exploitation of the possibilities provided by modern information technology, theories of learning must be incorporated into planning of the learning environment (Pohjolainen, S. et. al., 2001). Educational features should be taken into account in planning. Educational goals, methods, context, content and task analysis, target group analysis, training need analysis, feedback, tutoring, assessment, narrative, usability and competency requirements are key factors. Training scenarios and technological requirements could be developed after definition. The use of varied educational technologies without educational planning can hinder or even forbid learning processes.

When we are talking about using information and communication technology in education the concept educational technology is used. Educational technology is the systematic and creative blending of "idea" and "product" technologies with subject-matter content in order to engender and improve teaching and learning processes. Educational technology is often associated with the terms instructional technology or learning technology. (Wikipedia, 2006e.) Very common concepts are also eLearning and Online learning, which are mainly focused on the Internet-based applications.

Pohjolainen et. al. (2001) emphasise strong forms of interaction learning environment by designing the assignments, the content, the feedback, the collaborative and cognitive tools to operate together in supporting the learning towards the objectives. The learners are not acting impulsively or haphazardly in the learning environment, but are able to take cognitive responsibility of their own learning process. It is possible to reach deeper understanding by taking the cognitive responsibility and reflecting on what is to be expected (Lehtinen, 2000). However, one cannot guarantee that every learner takes the cognitive responsibility even if the methods and tools are "optimal".

Grabinger and Dunlap (Grabinger & Dunlap, 1995) emphases that learners should be presented with complex environments that represent interesting and motivating tasks, rather than contrived sterile problems (see also Harper, Squires and McDougall 2000). Appropriate feedback and meaningful interactivity are important parts of the learning process (Youngblut, 1997).

Pedagogical guidelines for designing computer-based learning environment for complex learning (Lehtinen, 2000):

1. Facing complexity from the beginning of the study process.
2. Appropriate integration of expert guidance and spontaneous exploration.
3. Integration of authentic problems and abstract reasoning.
4. Improving task-related social interaction.
5. Providing multiple representations.

In the context of rider training the appropriate approach should be defined by professional rider instructors. Without knowledge and experience of domain, it is not possible to develop appropriate solutions.

3.3 Simulator-Based Training

Interactive simulation is a special kind of physical simulation, often referred to as a human in the loop simulation, in which physical simulations include human operators, such as in a flight simulator. A computer simulation is an attempt to model a real-life situation on a computer so that it can be studied to see how the system works. By changing variables, predictions may be made about the behaviour of the system. (Wikipedia, 2007c.)

The theoretical and practical frameworks presented in this paper support the development of appropriate simulator based learning environment for real-life and game-based simulation. We should know the context as a whole before we can develop simulators, methods or materials. This makes it possible to create the direction and steps which will support the planning of the simulators. The effectiveness of the training is significantly influenced by, in addition to the simulator itself, the learning methods, the learning materials, and the guidance resources employed. (Ranta, 2005.)

Van Emmerik compiles factors of simulation environment. Learning in a simulation environment proceeds through interaction with (and manipulation of) a model. A model is a representation of (a part of) the real world in mathematical equations. Models are by definition a simplification of reality and as such, they can be used very well for training and instruction purposes.

Simulators has many added values e.g. repeatability, secure environment, economy, possibilities to control and plan practice, but also limitations (Vartiainen, 1985; Reed et. al., 1999; Hoeschen & Bekiaris, 2001).

The effectiveness of simulator training is significantly influenced by:

- Instruction, guidance and tutoring
- Resources
- Competence of a trainer
- Learning method
- Learning environment
- Motivation, learning style and previous knowledge of a student
- Feedback and reflection possibilities
- (Authentic) tasks and exercises
- Pedagogical features of a simulator

(cf. Vartiainen, 1985; van Emmerik, 2004; Ranta, 2005.)

According to Hoeschen and Bekiaris (Hoeschen & Bekiaris, 2001) a main advantage of the use of simulators is to train tasks which are too hazardous in real traffic. Indeed the simulators are an ideal way of training hazard perception without endangering the rider and other road users.

The opportunities to provide feedback in the synthetic environment are enhanced in comparison to reality (van Emmerik, 2004). A computer is an objective, all-seeing and tireless evaluator of ones performance. Also an appropriate feedback and meaningful interactivity are important parts of the learning process (Youngblut, 1997). These elements are connected to reflection processes, which are key elements to affect mental mind construction processes. We can reflect in or on action (Schön, 1983), if the simulator environment gives meaningful multimodal feedback. The context of the situation and trainees' level of competence makes it difficult to choose and develop reasonable feedback and activate reflection processes.

Educational and psychological factors are very important to take into account. For example complex, immersive and very fast phenomena can affect too much cognitive load, so the learning processes are overloaded or other processes require all the capacity (Sweller, 1988; cf. Ranta, 2003). This is not very simple question, because we are able to reflect our learning process after the action. The reflection processes are very important in learning, because it makes possible to change mental picture (schema) correspondent to real world. The key question is how we are able to model natural refection processes of a student so that it encourage reflect in and/or on action indirect or direct ways. There are varied instructional methods e.g. collaboration, tutoring, pedagogical illustrations, assessment of strategic, tactic and operative processes.

Transfer of learning is basic concepts of simulator based training. The learning processes and outcomes should be transfer to the real environment. Transfer of learning is the application of skills and knowledge learned in one context being applied in another context (Cormier & Hagman, 1987). De Corte (de Corte, 2003) defines transfer as the broad, productive, and supported use of acquired knowledge, skills, and motivations in new contexts and learning tasks. We can also use concept infer, which emphasis support of decision making. The interaction of a student and also the transfer of learning are possible to be developed by simulator-based training, because it serves safe, editable, repeatable, and flexible learning environment.

The interaction of a student and also the transfer of learning is possible to be developed by simulator-based training, because it serves safe, editable, repeatable, and flexible learning environment.

Artificial methods (AI) have been applied to adaptive hypermedia (AH) for at least 30 years now. At first, the attempt was to replace the domain expert with AI-based hypermedia application designed to give direct answers, guidance, or feedback to the user. Bareiss and Osgood. (Bareiss & Osgood, 1993) argue that the knowledge of a domain expert can be useful in more ways than many AI-based expert systems; expert can give user advice and describe relevant experiences and, most importantly, offer a rich context to interpret the advices and experiences. According to Woolf (Woolf, 1988) computer-aided teaching systems can be divided into weak and strong systems. A weak system is able to e.g. recognise insufficient answer or errors in user's actions. A strong system is a combination of an expert system and a teaching system. It is able to solve the problem in the same way as a domain expert and also guide the student when he is solving the problem. Designing and implementing a strong teaching system is much more complex compared to a weak system. (Woolf, 1988.)

Ranta (2005), van Emmerik (2005) and Nählinder (2006) find out possibilities e.g. to use pedagogical augmented virtual reality methods in virtual learning environment, which means that a trainee has a chance to get illustrations meaningful to learning, which cannot be used in real learning environments. In this way an expert or a teacher can analyse operation and they are able to make appropriate guiding decisions. This gives also possibility to create easier learning environments (context) for newcomers and several possibilities to support learning, scaffolding learning, debriefing and reflection in and on action.

According to Lehtinen effective simulated learning environment should fulfill some basic criteria (Lehtinen, 2000).

1. Simulated cases or problem situations have to be presented authentically. This means that students have to be able to believe that exactly similar problems can be faced in professional practice.
2. There must be enough representational support available for students, which make them possible to cope with the complex problem without too demanding cognitive load.
3. There should be flexible opportunities for students to move between concrete cases and systematic theoretical models.
4. There should be a feedback system in the environment which encourages students to reflect their own solution making processes.
5. The learning environment should facilitate student-to-student and student-to-teacher interaction.
6. The computer-based environment should be an integrated part of the whole instructional context.

Huhtamäki et. al. suggests, that there is also a need for a system, which will capture the recording of the action in a way that it can be reflected. The user is more likely to learn when his progress is tracked and meaningful information is delivered to him as he is experiencing problems. In the simulator domain the debriefing is a very important part of the learning and teaching process. If we can move a part of the debriefing to system using intelligent tutoring systems, it is possible to evaluate the recorded action according to a set of appropriate themes. (Huhtamäki et al., 2005.)

Illustrations of dynamic features of riding a motorcycle in different contexts support understanding and meaningful learning. Nowadays very authentic and augmented real time and dynamic simulations are possible to be modeled and simulated. These features are interlinked with understanding of very complex domain. The pedagogically sound dynamic illustrations have a plenty of added values to support construction process of knowledge.

In the development of e-Coaching based rider training varied features of educational framework are important to be taken into account. The focus and aims of a development project affects very much to the educational features and requirements of a system. Development of educational technological system needs several learning experiments to be done. When we are using game features the balance of edutainment should be evaluated carefully.

The road-safety research literature of the last two decades indicates a shift from training maneuvering skills to training higher-order skills, i.e. hazard perception and anticipation skills, risk perception, as well as self-assessment. It appeared that combining practical exercises, theory and self-evaluation of one's own behaviour may increase safety of novice drivers. A promising way to train cognitive skills are multimedia PCs and driving simulators. The main advantage of using these devices, instead of real-car and/or theoretical training, is that trainees can experience the consequences of their own decisions and actions in risk-inducing and hazardous scenarios, which is impossible on the real road. Trainees can develop these skills even before they have fully automated their maneuvering and vehicle control skills. The aim of such a novel training approach is accelerating the development of key safety skills that novice drivers lack due to inexperience, and to make young drivers aware of the effects of their personal conditions on actual driving behaviour. (Hoeschen & Bekiaris, 2001.)

3.4 Digital Game-Based Learning

Simulators are particularly attractive to the younger riders, since their lifelong exposure to digital media has raised them to be more receptive to this form of learning. According to Prensky (2001) “Within most of our lifetimes pretty much all learning will become truly learner-centered and fun — fun for students, fun for trainers and teachers, fun for parents, supervisors administrators and executives. The huge wall which has separated learning and fun, work and play for the last few hundred years is finally beginning to tremble and will soon come tumbling down, to everyone's benefit."

Prensky (2001) lists the following differences between the younger and older generations: The Games Generation has had far more experience at processing information quickly than its predecessors and is therefore better at it. They are more used to take in information in less sequential manner from multiple sources and creating their own structures and patterns. They are accustomed to get information in visual form with little or no accompanying text, thus sharpening their visual sensitivity and making it more natural for them to receive information this way. Finally, growing up with computers has engendered an overall attitude toward technology in the minds of the Games Generation that is very different from that of their predecessors.

According to literature, cognitive skills are best exercised by utilizing filmed clips, videos or digital media, where the trainee has to detect certain cues to predict what could happen and what he/she would do. Trainees should also be given comprehensive feedback. With simulators, trainees train cognitive skills without fully automated maneuvering skills. Simulators enable repeating certain scenes in changing environments, thus providing an excellent opportunity to understand how cues in traffic and outcome are related. Moreover trainees can experience the results of their own risky choices. With simulators it is possible to train anticipating skills, like hazard perception, which are highlighted by recent research as very important for safe driving. (Hoeschen & Bekiaris, 2001.)

Digital game-based learning e.g. game sport simulators like MotoGP, virtual living environments like Sims®, strategic war games and educational games are very familiar at many homes. It is possible to call these types of games edutainment because they combine education and entertainment (Wikipedia, 2007). Edutainment (also educational entertainment or entertainment-education) is a form of entertainment designed to educate as well as to amuse. Edutainment typically seeks to instruct or socialize its audience by embedding lessons in some familiar form of entertainment: television programs, computer and video games, films, music, websites, multimedia software, etc.

According to Wikipedia, most often edutainment seeks either to tutor in one or more specific subjects, or to change behaviour by engendering specific sociocultural attitudes. This is also used for behavior students in certain schools in the UK. Successful edutainment is discernible by the fact that learning becomes fun and teachers or speakers educate an audience in a manner which is both engaging and amusing. The field of Entertainment-Education (EE), combines communication and education theory with communicative arts to deliver primarily social development messages. (Wikipedia, 2007b.)

The game-based learning approach is some kind of indirect learning method, which stresses interior motivation, emotions, natural way of people to use free time and compete with each other. Indirect ways has a lot of connection with the ideas of constructivism, where student centered methods are important (cf. Jonassen, 1995).

It is not easy to design an educational game-based learning environment, because there are challenges to utilise appropriate game features in deep connection with educational objectives. If the design principles stress too much education, some people can think that this is not motivational. On the other hand too much entertainment can even guide the learning to the wrong direction. These features are also context and situation dependent. So it is necessary to use common concepts, motivating narratives, meaningful scenarios, reasonable amount of decision making tools, illustrative indicators of performance, usable ways to navigate and supportive reflection tools. Designing a game-based learning environment requires a multidisciplinary team, where professionals of different domains are involved in a collaborative way.

3.5 An IRT e-Coaching approach

In considering the psychological requirements and the opportunities offered by e-Learning, simulation and games based approaches, we are of the view that what we believe would be a new approach is required for the IRT e-Coaching programme. This would combine elements of “believable” simulation with gaming characteristics, including artificial intelligence and the requirement for “belief suspension”. Importantly it would be dynamic and require the user to make strategic decisions and understand their consequences in the context of identifying and managing hazards in traffic.

4. THE EVALUATION PROCESS AND METHODS

The Supervisory Board of the IRT Project decided in Paris on October 2005 to evaluate e-Coaching solutions for initial motorcycle riding. The IRT Supervisory Board, as a consequence of the contributions of Board member Mr. Pekka Ranta, agreed that the Hypermedia Laboratory would be asked to undertake an evaluation within the IRT Project of e-Coaching solutions to be used on PC and/or video game consoles.

This document is a product of an interactive and iterative process composed of cycles of research, design and discussion. After an initial research of the subject, a rudimentary plan for the e-Coaching approach was sketched out. This was then reviewed and developed with the IRT Project Co-ordinator in Tampere on September 2006.

Based on the observations made in this meeting, the scope of the study was refined. More research on the field of e-Coaching was carried out. Use cases were used to chart the needs of the future users. Furthermore, a few scenarios (see appendix E) within the e-Coaching programme and their contents were created to inspire more discussion in the next meeting, which was held in Brussels on November 2006. The scenarios were successful in creating a debate about the requirements and features of the e-Coaching programme, as well as forming a unified concept for the e-Coaching approach as a whole.

Using the list of requirements coined, a suggestion of the architectural design of the system was created and iteratively improved until it met all the requirements imposed. After this the alternative implementation technologies were studied and compared, and finally a rough cost estimation of the implementation was made based on the proposed design and technologies.

The process leant on constructive pedagogical framework, which quintessentially builds new reality based on the existing knowledge (Järvinen, 2000). In this case the existing knowledge can be divided into three sections: Firstly, the expertise of the motorcycle domain and the needs of the initial riders, provided by the IRT Project, Mr. Tomlins and colleagues. Second section is the knowledge of pedagogical aspects of the virtual learning environments and simulators, and the third part is the general understanding of computer technology.

The nature of this document is a recommendation. It is based solely on theory and previous research of the topic. Therefore it is crucial to create prototypes and technical tests before the final implementation of simulator software is undertaken. If the technical recommendations made here are found impractical, or better alternatives are discovered, they can be used in the place of the ones suggested in this document.

5. REQUIREMENTS FOR THE IRT E-COACHING PROGRAMME

The goals set for the IRT e-Coaching approach can be fulfilled by satisfying the following functional requirements. The requirements have been divided into multiple categories and numbered to enable an easy reference later on in the document.

5.1 Requirements for the Exercises

A1. The user can select an exercise addressing a specific aspect of the rider training from a list of predetermined exercises.

A2. The exercise themes are: positioning (at traffic speed and at slower than traffic speed), distance, riding on curves and bends (both left and right hand), anticipation (of other road users and of environment and infrastructure), junctions, overtaking, motorways, group riding and journey planning.

A3. A level-based approach would require the user to reach a standard before moving on the next level. As the difficulty of the level increases, so does its duration, the incidents of the random events, the choice of the alternative routes, the intensity and mix of other traffic and the effect of weather conditions.

A4. The level-based mode has a storyline or plot, which gives the user a motivation and reason for riding.

A5. Between the levels there will be optional bonus games that test the user's knowledge in matters related to the exercises. For example a bonus game might be a picture of a riding scene with multiple hazards, and the user has to identify them and rank them by priority.

A6. The system includes a introductory tutorial, which accustoms the user to the controls, scoring principles etc.

A7. The content of the IRT e-Coaching programme would be separated from the application logic, so that the content can be easily updated.

A8. The IRT e-Coaching programme would enable traffic movement on either left of right side of the road to be selected.

5.2 Requirements for Controlling the IRT e-Coaching Programme Tasks

B1. The methods for controlling the tasks must be realistic enough to enable the user to identify his/her actions with riding a motorcycle in real-life.

B2. The user can control the speed of the bike by accelerating or decelerating.

B3. The user can control the direction of the bike by turning to left or right.

B4. The user also controls the turn signals and the horn.

B5. The system automatically calculates the gear used based on the speed and acceleration. Hence the user doesn't control the gears and can concentrate more on hazard perception.

B6. The user can alter the view angle to both sides, as to mimic the action of turning ones head and looking to the sides.

B7. The mirrors of the bike are blurred until the user tells the system, that he/she is paying attention to the mirrors by clicking on them. After a click the mirrors remain clear for a short period of time, showing the user in a realistic manner what is happening in the traffic behind.

5.3 Requirements for Evaluating the Performance of the Trainee

C1. The system has a reference model that defines the parameters of safe riding. These parameters are the speed of the motorcycle, distance to other objects and visibility.

C2. The system can measure the user's performance on safe riding by measuring the values of the aforementioned parameters.

C3. The system can compare the results of the performance measurements to the reference models of safe riding.

C4. Based on the aforementioned comparison, the system can provide the user with feedback in different forms:

- Numeric feedback (score and statistics).

- Textual feedback, which can be direct (e.g. ”You crashed because you didn't check your mirrors.”) or indirect (e.g. ”You might want to pay more attention to traffic.”).
- Graphic feedback (replay of the user's performance).

C5. The system has a list of relevant events. When a certain event is fulfilled, a trigger goes off and causes predefined actions.

C6. By measuring the parameters of safe riding and using the event-trigger pairs the system recognises when the user is deliberately using the IRT e-Coaching programme as a crash game, rather than trying to learn.

C7. By measuring the parameters of safe riding and using the event-trigger pairs the system recognises serious offences (e.g. disobeying the red traffic lights) and can interrupt the exercise in such cases.

C8. The system models the concept of a safety bubble. The safety bubble represents the amount of free space on the different sides of the bike, thus measuring the amount of leeway the user has. This information can be used to determine how safely or dangerously the user is riding at any given point of time. The system can dynamically calculate, update and store the dimensions of the safety bubble.

C9. The system has a database of real-life accidents with the data of related injury to the rider and approximated cost to repair the bike.

C10. The system can evaluate and categorise the possible in-game accidents.

C11. By comparing the database of the real-life accidents and the results of the evaluation of the simulated accident, the system can report the most likely effects of the crash for the rider (injury) and for the bike (cost to repair).

C12. The system is capable of calculating the breakaway point (the moment when the user loses the control of the bike) based on the inputs of the user and the prevailing conditions. The model used in these calculations is based on an inexperienced rider, making the breakaway point relatively low.

C13. The system can calculate whether or not acceleration is intense enough to aviate the front wheel of the bike, and animate this to the user if necessary.

5.4 Requirements for the Debriefing

D1. A recording of the user's performance can be replayed from different view angles (e.g. the bird perspective, from other vehicles, from an imaginary training instructor's point of view) in the debriefing phase.

D2. The system incorporates selection and time control tools. The selection tools can be used to mark hazards in a single frame created from the riding mode with the time controlling tools.

5.5 Requirements for the User Interface

E1. The user interface in the riding mode is kept simple to create a realistic riding sensation. In addition to the visible parts of the bike (the handlebar), the user interface contains as few augmented user interface elements as possible.

E2. The handlebar of the bike contains information of the current speed and gear. The milometer represents the user's progress in the current level or exercise.

E3. The handlebar and the riders hands are animated and realistically perform the motions of controlling the bike, e.g. braking and changing the gear.

E4. The user can select whether or not the current score is shown while in riding mode.

E5. The system must be implemented in a way that enables localisation of texts throughout the whole system.

E6. The user can select whether to use miles or kilometers on the speedometer and traffic signs.

E7. The user can navigate through the programme by selecting options from menus.

5.6 Requirements for the Graphics and Views

F1. The surrounding environment is created using simplified and general, cartoon-like graphics. However, the graphics must be realistic enough to enable an easy transference of learned things into real world.

F2. When the user is riding in bends, the world gets tilted while the bike remains in the upright position to create a realistic visual representation of the bike tilting.

5.7 Requirements for the Sounds

G1. As in the real life motorcycle riding, the sound environment is limited by the muffling effect of the helmet. The user only hears the sound of the motorcycles engine and loud noises like a car horn or siren of a rescue vehicle.

5.8 Requirements for Identifying and Monitoring the Users

H1. The users can be authenticated by username and password as they log in.

H2. If the user misbehaves, he/she may temporarily lose his/her driving license, which prevents him/her from logging in for a short period of time.

5.9 Requirements for the Communal Collaboration

I1. User can publish his/her score to a top score list available for everyone.

I2. User can challenge peers to take a certain exercise.

I3. User can compare his/her performance and statistics with those of his/her peers.

I4. User can discuss the exercises and other aspects of the programme with peers.

6. TECHNICAL FEATURES OF THE PLATFORMS

This chapter reviews the PCs and video game consoles as platforms for the designed IRT e-Coaching programme. In computing, a platform describes some sort of framework, either in hardware or software, which allows software to run. Typical platforms include a computer's architecture, operating system, or programming languages and their runtime libraries. (Wikipedia, 2006b.)

6.1 Advantages and Disadvantages of the PCs

The most significant advantage of the regular PCs is the widespread and vast device base. At the end of the year 2005 there are over 900 million PCs in use worldwide and it is estimated that the number of new units sold 2006 through 2010 might be as high as 1300 million (CIA, 2006a; CIA, 2006b). Since the PCs have many other functions in our everyday life and are affordable to acquire, they have become widely available at homes across Europe. In addition many public (schools, libraries) and commercial (Internet cafes) organisations offer an access to PCs.

Nowadays the PCs are common appliances, and therefore the users are familiar with their operating systems (OS), user interfaces and operations models. Especially for the younger generations the usage of computers comes as a second nature. The controllers (keyboard and mouse) are rather clumsy for controlling a motorcycle, but since the machine controls are not a central part of this programme, the controllers are likely to be adequate. However the IRT e-Coaching programme could be designed to be used with a readily available joystick device, subject to the PC´s operating software.

Using PCs also enables a more efficient utilization of Internet connections as a part of the IRT e-Coaching programme. This makes it possible to easily distribute and update the content. Social aspects, such as discussions and challenging friends, can be integrated to be a part of the IRT e-Coaching programme and implemented via Internet. This can be seen as a competitive advantage against the video games with cutting-edge graphics. Storing user data on a web server makes it possible for the user to resume his/her previous training session on another computer, thus relieving the user from the restriction of training, for example, solely on the home computer.

The PCs naturally have some disadvantages as well. A PC may contain virtually an infinite number of different component configurations. Therefore the IRT e-Coaching programme can't be designed for the most powerful PC system available, as this would make it impossible to run it on an average PC. The problem isn't limited only to the hardware: the installed software may also vary greatly.

6.2 PC Environment and Interfaces

6.2.1 Minimum System Requirements

In order to be able to make a proper evaluation about the best implementation technologies, the minimum specifications of the PC system must be appointed. As new, more powerful PCs are rapidly developed, this decision is based on what can be estimated to be a commonly available system two years hence.

At the end of the year 2006, the following or equivalent PC system can be bought for approximately 500€:

- 3GHz CPU
- 512 MB or 1024 MB of RAM
- Relatively powerful, non-integrated GPU

It is quite safe to assume that such system will be widely available at the time of the completion of the IRT e-Coaching programme.

6.2.2 Software Environment

As the different versions of Microsoft® Windows® are dominating the operating system markets, they are an obvious choice as the primary target OS. However, if it is relatively easy to make the teh IRT e-Coaching programme a cross-platform product (i.e. supporting multiple operating systems), this would be recommended so that the programme would be available to maximum audience without limitations on the operating system. For example using Microsoft's DirectX libraries will limit the software to Windows environment, but cross-platform alternatives for these libraries are available.

An Internet browser will be required to access the programme distributed via Internet. As Microsoft® Internet Explorer is distributed with the operating system, and other browsers can be downloaded from the Internet free of charge, availability of a browser is not a problem. However, the website and the e-Coaching programme has to be designed the multitude of different browsers in mind, in order to be able to offer an equal use experience for the users of the different browsers.

In order to run Java-based applications, the users must install Java Runtime Environment (JRE) on their computers. JRE is free software by Sun Microsystems. It can be freely downloaded from their website. Although the installation process is quite simple, it may cause problems to some users, particularly to those with insufficient user rights to install new software. Depending on the implementation technology, other software, like web browser plug-ins, may be required. However unnecessary plug-in dependencies should be avoided.

6.3 Advantages and Disadvantages of the Video Game Consoles

Although video game consoles are quite widespread, they cannot match the PCs in numbers. According to Sony® (Sony, 2006), it has sold 42 million PlayStation® 2 consoles to the European market. Microsoft states (Microsoft, 2006) the number of Xbox® consoles sold worldwide to be 24 million. It is estimated, that Nintendo's GameCube has even lesser sales numbers. To reach even these video game consoles, three separated versions of the game would have to be made. It is unlikely, and against the goals of the project that the users would acquire a specific video game console just to use the simulation software.

The sixth generation video game consoles (PlayStation 2, Xbox and GameCube) were published at the start of the decade (2000, 2001 and 2001, respectively). Consequently the hardware used in them is relatively old. However, as each unit of one console type has exactly the same hardware configuration, it is easier to optimise the software and utilise every last bit of computing power. The seventh video game console generation has significantly better hardware, but at the end of 2006 it is too early to estimate how widespread they will become.

Microsoft offers a game developing toolkit called XNA's Not Acronymed (XNA). Current version of XNA is intended for hobbyist developers who can create their own games for Microsoft Windows XP and Xbox 360 consoles and share their creations with other users. This however requires that each user who wants to run their games on the Xbox 360 will have to pay an annual fee of $99 for admission to the XNA "Creators Club". Each user also needs a Xbox 360 with a hard drive, which is not available on the cheaper version of the console, Xbox 360 Core System. Distribution of the games is rather troublesome, because the users themselves need to compile and deploy the software. (XNA FAQ, 2006.)

A professional version of XNA is scheduled for release in 2007. Although the current version is unsuitable for the IRT project due to its unpractical nature, the upcoming professional version might be interesting as a ready development toolkit might cut down the time required for development.

The PlayStation 3 development toolkit is based on open and publicly available APIs. Sony has selected several technologies and arranged several sublicensing agreements to create an advanced software development kit for developers. Sublicensed technology includes complete game engines, physics libraries, and special libraries. (Wikipedia, 2006d.) If PlayStation 3 succeeds in becoming a popular console despite its relatively high price, this development kit makes it an interesting option for the IRT e-Coaching programme development.

Nintendo hasn’t published information concerning publicly available game development toolkits for its seventh generation console, Nintendo Wii.

A major inconveniency with video game consoles are the royalties, which must be paid for the console manufacturers when a new game is developed.

6.4 Advantages and Disadvantages of the Internet-Based Approach

A notable present trend in software development is the utilisation of the Internet, which offers considerable additional value for the IRT e-Coaching programme project as well. Not least of these benefits are the open web standards. For example World Wide Web Consortium (W3C) strives for creating a set of recommendations, which will ensure the interoperability of different systems now and in the future.

The software can be distributed at any time to any place economically via the website, eliminating the need for installation media like CDs and the problems associated with them, like manufacturing, distribution etc. If CDs were used, every new version of the software (patch or content update) would require a new batch of discs to be distributed all over Europe.

The bandwidth of the Internet connection can easily become a bottleneck. In order to sensible use the Internet as a distribution channel the users are required to possess a broadband Internet connection. Even the slowest broadband connections may turn out to be a challenge for smooth operation. However, this should not pose an accessibility hazard to European users, as according to the OECD, the number of broadband connections in EU15 countries have been quickly rising, totaling 5.9 connections per 100 inhabitants in 2003, 9.7 in 2004 and 14.2 in 2005 (OECD, 2006). The advancements in technology and the declining prices increase the speed of the current connections across Europe.

The user data can be stored on the server database. This way the users can resume their training session on different computer, as the state of the user can be downloaded from the database and saved again at the end of the session. This is especially suited for the users, who use the IRT e-Coaching programme on public computers, such as the ones in libraries or schools.

Another possible bottleneck is the server system. Especially when the server is under heavy workload produced by numerous users, considerable processing capacity is required. Another challenge is keeping the round-trip time, i.e. the time it takes the server to reply to a client software, within acceptable limits. These problems can be partially solved by making the client software as autonomous as possible.

Internet also provides means for communal collaboration. This can be seen as a remarkable asset when trying to attract the young gamers. Upkeeping a top score list and giving the users a change to compare their performance may create a constructive competition between the users. The users can easily challenge their real-life friends to take the same exercises, thus attracting new users. Providing a venue for a discussion about safe motorcycle riding helps the users understand why they got a poor score on the IRT e-Coaching programme, and can cover the related topics not covered by the programme itself.

The Internet-based approach requires a server. Since the IRT e-Coaching programme has such a wide user base, multiple servers are recommended: a separated server for hosting the website of the the IRT e-Coaching programme, and another dedicated server for performing the operations required by the numerous simulator instances running on users computers. To provide reliable downloads and speed up the downloading of the IRT e-Coaching programme, multiple mirrors (servers containing identical copy of the files) can be utilised.

6.5 Conclusion

Since the PCs are today common household appliances, they offer wider accessibility for the young initial riders than the video game console solutions. To reach the different video game consoles, multiple versions of the software would be required, each producing additional costs in the form of royalties. This would also tie the project to the decisions of the respective companies, whereas the standardized PC-based solutions are not dependent on the whims of a single vendor.

Due to technical limitations, the Internet-based approach can be fully utilised only on the PC platform. Furthermore, for the PCs there is a multitude of tools and existing components, which can be utilised in this project. These, and other advantages the PCs have, strongly suggest that the IRT e-Coaching programme should be implemented as a PC-based software.

7. PRELIMINARY DESIGN AND IMPLEMENTATION TECHNOLOGIES OF THE IRT E-COACHING PROGRAMME

This chapter makes a preliminary sketch of the synopsis and the user interface of the IRT e-Coaching programme. It also presents a preliminary proposal for the general architectural design of the system and defines which components corresponds to the requirements presented on chapter 5. To clarify how the proposed system would function, a process description is also provided. Finally the advantages and disadvantages of the alternatives for implementation technologies are considered.

7.1 Synopsis

The user experience of the IRT e-Coaching programme begins at the website of the programme. On the website the user can register a nickname, log in using it and launch the programme. The website also contains an area for discussion on motorcycle related topics and a top list of best performances. If the user has been challenged by other user to take a certain exercise, a notification is displayed when the user logs in.

The IRT e-Coaching programme itself contains two main sections: a level-based mode and an exercise-based mode. In the level based mode the user completes levels with random events and increasing difficulty. The first level is rather easy, but as the level increases, the hazardous situations get more frequent and random, more alternative riding routes are offered, and the time required to complete the level ascends. In the final level the hazardous situations are frequent and more than one may occur simultaneously. Between the levels there are small optional bonus games.

In the exercise-based approach the user may select which area of the motorcycle riding he/she wants to practise. For example, if the user wishes to practise overtaking, the exercise will contain many overtaking situations, but little or no junction related situations.

While riding, the performance of the user is constantly monitored. By measuring the speed, the distance to other objects and the visibility, it is determined how safely the user is riding. Safe riding grants points, while reckless riding deducts them.

The IRT e-Coaching programme contains a mentor character, who is a renowned rider, such as Valentino Rossi. If the user is having problems, the Mentor will offer help. If the user crashes the bike, the Mentor will tell the likely consequences that such a crash would have in the real life. To emphasise the consequences, the feedback could be given in a hospital or repair shop environment.

Before a level or an exercise, a briefing is held. In the briefing the Mentor sets the background for the upcoming exercise by telling the user what he/she is supposed to do. After a level or an exercise, a debriefing is held. Here the Mentor praises the user for the things that went well, and offers tips for the parts the user had difficulties with. The user can also view a recording of his/her latest performance, submit the score to the top list on the website or challenge a friend to try the same exercise and top the user's score.

If it is obvious that the user is using the IRT e-Coaching programme just as a crash game, rather than benefiting the experience, the programme will inform the user, that he/she has temporarily lost his/her drivers license. It will then shut down, and the user can't launch it again before a predefined amount of time has passed.

7.2 User Interface

The user interface can be roughly divided into four sections: the menus, the briefing, the programme and the debriefing. Each of these sections has their own purpose, and therefore designing each section has a different focus. The website hosting the IRT e-Coaching programme can also be seen as a part of it, and therefore it is important to define the things that need to be considered while designing the website.

As stated in the requirement E5, the whole system has to be designed and implemented so that all the texts can be easily localised to different European languages. This includes menu texts, information in briefing and debriefing screens, suitable areas of the programme itself and the website.

7.2.1 The Menus

The user navigates the system using menus. This includes, for example, selecting the starting level of the level-based approach or configuring the in-game options.

Although the menus play a minor role and are quite trivial to design, it is important to ensure that they do not hinder the usage of the system, especially for the users who are less experienced with computer systems. Good way to minimize the problems the users might face with the menus is to use a convention the users are already familiar with (from using other software). On the other hand, if a suitable metaphor from the current domain (motorcycles) is found, it could be used instead.

7.2.2 The Briefing

The briefing screens offer a short description of the exercise at hand, instructions, and options to either start the exercise or return to the previous menu. The briefing screen provides the user only with the instructions required for completing the task at hand. General instructions that apply throughout the programme are given in a separate tutorial.

The purpose of the briefing is to give the user an objective: why and where is the user riding, what is the user trying to achieve. The purpose is not to give the user unnecessary hints about the exercise itself, so the instructions are kept to minimum.

7.2.3 The IRT e-Coaching Programme

The parts of the programme that contain the actual riding experience are kept as realistic as possible, as mentioned in the requirement E1. To compensate the lack of augmented UI elements, the operations of the system are executed as metaphors when possible. For example pausing the game is executed by slowing down and stopping at the side of the road instead of pressing a button. Quitting the exercise is done by turning off the engine of the bike instead of selecting ”quit” from a menu. If the user needs to find a specific location, he/she needs to follow the authentic signposts scattered around the city, rather than look at a separate map function.

Basic information is given to the user in the form of a handlebar of the motorcycle, as this will make the riding experience more realistic than detached meters hovering on the corners of the screen. The handlebar contains information of the current speed (speedometer), gear (gear indicator) and progress (milometer).

The handlebar and the rider's hands are animated. When the user accelerates, the hands perform realistic actions and the reading on the speedometer rises. When the speed rises past a predefined point, the gear is automatically changed with the hands performing the necessary actions and the reading on the gear indicator changing. Braking which is under the controller of the user is shown on the screen in a similar way.

7.2.4 The Controls

The most important control the user has over the motorcycle is the speed, which in turn controls distances and leeway. While riding, a single keypress increases/decreases the speed slightly, while repeated tapping of the button changes the speed at the rate dependent on the frequency of the tapping. A long keypress causes a fierce acceleration or an emergency braking. Without user input the speed remains constant.

The user can control the direction the bike is heading by steering. The longer the user taps the button, the longer the turn lasts. Faster tapping causes tighter turns whereas slower pace leads to wider turns. That is to say that short but rapid tapping results in quick turn (evasive move in emergencies) and long, unhurried tapping makes the bike to ride in a circle with a large radius.

User can also turn the view by turning the head of the rider. By pressing the button, the head begins to turn to the side, until the head is turned as much as realistically is possible. The head remains in this position until the key is released, at which point the head quickly returns to the centre position. Furthermore the user has a control over the turning signals (on/off) and the signal horn.

As previously stated however the IRT e-Coaching programme could be designed to be used with a readily available joystick device, subject to the PC´s operating software.

7.2.5 The Debriefing

The debriefing is important part of the learning experience, as the feedback given after the exercise might reveal things that the user might have overlooked during the exercise.

The most essential content of the debriefing is the feedback given by the Mentor. This feedback should be given in a form that the user can and will understand and accept. Therefore the text has to be relatively straightforward, so that the users understand it, and short, so that the users won't skip over it.

7.2.6 The Debriefing after an Accident

One goal of the IRT e-Coaching programme is to teach the riders the consequences of incorrect actions and attitude in traffic. This is achieved by telling the users how bad injuries and monetary losses the accidents would cause them.

If the accident would likely cause severe injuries to the rider, the debriefing is held in a hospital location. Based on data collected from real-life accidents, the Mentor will cover the injuries that the rider would be likely to suffer in such accident. The repair cost of the motorcycle is also mentioned.

If the accident would likely cause only minor injury to the rider but damage the bike, the debriefing is held in a repair shop location. Based on the real-life data, the cost to repair the motorcycle is reported.

After an accident the debriefing screen offers the options to view a replay and either to re-ride the exercise or level or to return to the main menu.

7.2.7 The Debriefing after a Successful Exercise

After a successful exercise the user receives a brief praise from the Mentor. If the user has room for improvement on any sector of safe riding, the Mentor hints about this, but the user can continue to the next level or exercise anyway. On the exercise-based mode there is a brief review of the covered subject.

After a successful exercise the debriefing screen offers the options to view a replay, play the bonus game (on the level-based mode) and either to continue to next the exercise/level or to return to the main menu.

7.2.8 The Exercise Replay

In the debriefing phase the user can replay a recording of the performance in the preceding exercise. Augmented reality cues, such as the distances to other vehicles can be added into this recording. The replays can be viewed from different camera angles.

The user can choose to view the replay from the following angles (in addition to the regular first person view): Bird perspective shows the bike from behind, few meters high from the ground. This view can be useful while going over close call situations. Other traffic view shows the bike from inside of the other vehicles or the pedestrians nearby, thus demonstrating how hard it is for others to detect the rider. One option could be a driving test view, which would show the replay from the eyes of motorcycle driving test examiner, so that the user can evaluate how an examiner would have seen the user's performance. The user can change the view angle on the fly and control the replay like a video: play, pause, fast forward and backward.

There are many useful virtual reality augmentations for the IRT e-Coaching programme. An explicit display of the distances to other objects and the room required for bringing the bike to halt can aid the user to determine correct positioning. This can also be achieved by highlighting the objects that the user gets too near to. A visible trail of the bike can reveal close call situations to the user.

Combining replays from different view angles with augmented reality can offer the user whole new perspectives on the actions taken during the exercise and thus reveal hazards that remain hidden in the first person view. Such revelations can help the user to take these hazards into account later on, on the real life.

7.2.9 The Bonus Games

The bonus games occur between the levels of the level-based simulation, offering the user a change to gain bonus points. The bonus games are technically simple point-and-click exercises. The look and feel of the bonus games should be as close to the rest of programme as possible.

7.2.10 The Website

If the IRT e-Coaching programme is launched from a website, it will be the first step the user takes, thus giving the first impression. The website has to be designed and produced the common usability and accessibility principles in mind, so that it won't hinder the usage of the programme.

The general appearance of the website should be consistent with that of the IRT e-Coaching programme, forming a uniform entity. The website should contain, in addition to the programme itself, medium for the social aspects of the programme and guidance for possible problems the user might face.

7.3 Architectural Design

The IRT e-Coaching programme is composed of the following main components: The e-Coaching Manager, the Exercise Manager, the Communications Manager, mathematical models and data storages. The components, their functions and interactions are described below. An illustration of the proposed architectural design is given in the appendix C.

7.3.1 e-Coaching Manager

The e-Coaching Manager forms the core of the whole system and interacts directly with the other components. It is responsible for executing the exercises created by the Exercise Manager. During the execution of exercises the e-Coaching Manager conducts the operation by telling other components what to do. The e-Coaching is closely connected to the Performance Evaluator, the Instructor and the Internal Object Data Model (IODM).

Jain and McLean (Jain & McLean, 2004) define the Internal Object Data Model as a data model, which ”is used to maintain the status of all the objects in the scenario”. In the IRT e-Coaching programme it is a database that keeps track of all the objects (e.g. vehicles and pedestrians) in the riding mode by constantly updating their locations, directions and other attributes. This database is ”runtime”, i.e. it is strictly tied to the current exercise: the database is populated in the start of the exercise, updated during it and cleared when the exercise ends. In other words, no data is stored in this database between the different exercises. The IODM communicates solely with the e-Coaching manager, which either tells the IODM to change an attribute value of an object or inquires objects current attribute values.

The Performance Evaluator gets information about the user's actions from the e-Coaching manager. It uses this information to calculate the dimensions of the safety bubble and monitor whether or not the user's actions causes one or more of the predefined event triggers to go off. Also other aspects of safe riding, such as the usage of turning signals etc. are measured to evaluate how safely the user is riding, and which fields of the user’s performance still have room for improvement. This evaluation is relayed on to the Instructor.

In case of an accident the Performance Evaluator compares the situation to the Crash Model, which is a parametrisation of motorcycle accidents. As a result the Performance Evaluator categorises the accident, so that it can be compared to a database of real-life accidents and thus report the most likely effects of the crash for the rider and for the bike. This information is also relayed on to the Instructor.

The Instructor gets its inputs from the Performance Evaluator. It creates a variety of feedback which it outputs to the e-Coaching manager. Numeric feedback is for example the score the user got on the level or the percentage of the time the user had a sound safety bubble. Textual feedback can be a description of the accidents consequences, a praise for a well performed exercise or hints to gain a better score next time.

7.3.2 Exercise Manager

The Exercise Manager reads an exercise definition file and parses different resources into an exercise package, which is then executed by e-Coaching manager. The resources that need to be gathered for a package include graphic and sound files, the environmental model, the rule base and user-specific information such as the side of traffic.

7.3.3 Communications Manager

The Communications Manager answers for communicating with the user and the server. User communications consists of user inputs and visual and audio outputs. Telecommunications relays messages between the IRT e-Coaching programme and the web servers.

The user's inputs are read from the keyboard and interpreted by the User Input Handler. If the input corresponds to a valid command, it is passed on to the Communications Manager, which will launch the required operations. The results of the operations, such as the calculations by mathematical models, will be relayed back to Visual and Audio Outputs. The Visual Output updates the screen and the Audio Output creates the noises for the user.

The visualisations in the IRT e-Coaching programme are created in a process called rendering. First it performs calculations, like rotating and scaling, to the 3D models of the objects. When all the calculations are done, it combines the different objects into a scene, which is then rendered into a 2D image. This calculation process is rather heavy, because it has to be done in real time, at the rate of approximately 30 images, or frames, per second. Since one goal of the IRT e-Coaching programme is to function on inexpensive hardware, the workload can be lightened by keeping the 3D models relatively simple.

The rendering process can be done in a multitude of ways. As non-photorealistic, cartoon-like graphics have been identified as possible approach for the programme, cel-shading is an interesting technique. It creates hand-drawn, comic book style graphics. Although the results of this approach may have simplistic look and feel, the process beneath is complex, and therefore its suitability for a light-weight the IRT e-Coaching programme has to be tested before utilisation.

By relaying the messages moving between the IRT e-Coaching programme and the web server, Telecommunications enables the social aspects of the programme. The users can post their scores to a top score list on the website, challenge their friends to take a particular exercise and beat their score, or discuss topics about the programme and motorcycle riding in general. By offering this additional value to the users the programme can compete with the graphically more advanced video games.

7.3.4 Models

A model is a representation of (a part of) the real world in mathematical equations. Models are by definition a simplification of reality and as such, they can be used very well for training and instruction purposes. (van Emmerik, 2004.) Generally the model is resolved with a computer and a mathematical solution is obtained. This is known as simulation. The mathematical solution is compared to the real world case. At this point validation of the model is also performed – an estimation of how good the model is and how reliable the results are. The simulation results obtained from the model can be compared to the measurements obtained from a real system. If the mathematical model is not good enough it can be improved by including more relevant factors in an iterative manner. (Pohjolainen et al., 2003.)

A lot of work has been done in the area of modeling a motorcycle. Most of this work has been carried out by motorcycle manufacturers, who use modeling extensively as a part of their product development. Nevertheless, modeling a motorcycle is far from a routine task. The biggest challenge for the IRT e-Coaching programme is finding a balance: The model has to be detailed enough to support the learning process, while being light-weight solution that won't hinder the performance of the system.

The dynamic model of the motorcycle calculates the reactions of the motorbike based on the user inputs and the environmental variables. These calculations can, for example, tell the programme how long after the start of the braking the motorcycle has come to a full stop with the current speed and weather conditions. Modeling the dynamics of the motorcycle is a challenging task, and enough resources should be directed for it since the quality of the whole programme greatly depends on it.

Since the purpose of the programme is to train motorcycle riders, a quite detailed model of a motorcycle is required. However, the emphasis of this IRT e-Coaching programme is on the hazard perception and rider attitude, and therefore modeling of all the minute details is not necessary. Foundation of the model is a midrange sporty bike with a motor of about 650cc.

In order to produce a functional, educational motorcycle riding programme, it is not sufficient to model merely the dynamics of the motorcycle. Other objects in the programme, such as the other vehicles and pedestrians, also need their own dynamic models.

However, these models can be much simpler than the one of the motorcycle, since the users are not likely to pay too much attention to their dynamic behaviour as long as it is substantially realistic.

One part of defining how safely the user is riding is the model of the safety bubble. The safety bubble represents the amount of leeway the user has on the different sides of the bike. The size of the safety bubble is calculated based on the speed of the bike and the distance to other objects on all sides. When the size of the bubble gets smaller, the speed of the bike has to be reduced to maintain the same level of safety.

In figure 7.1 we can see a demonstration of the safety bubble. On the left side the rider has a nice safety bubble and is riding ideally near the center of it. On right side the rider is passing a parked car while there is an oncoming vehicle on the another lane. Even though one of the cars is parked, situation like this poses a high risk, which is expressed by the shrinking safety bubble. Therefore the situation on the right hand side can be viewed as more dangerous than the one on the left hand side.

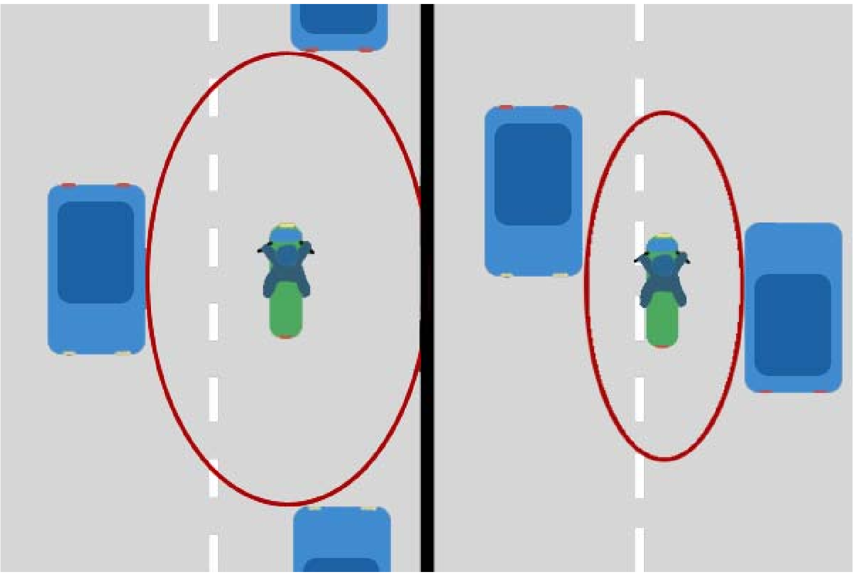


Figure 7.1: The safety bubble.

Crash model is used to categorise the crashes the user might have while riding in the IRT e-Coaching programme. The Performance Evaluator compares the crash conditions, such as the speed of the bike and the object the bike crashed into, with the crash model. Based on this comparison, the crash the user had is placed into one of the categories. By comparing this information to the database of motorcycle accidents, the likely consequences of the crash can be reported to the user.

7.3.5 Data Storages

The data storages contain the relatively static information of the system. Relatively static means that some data might be updated as a result of user's actions. The user data and accidents are stored in a form of databases, while graphics, sounds, exercises and rule base are files.

The user database contains the states of all the users: usernames, passwords, nationality, setting configurations, results of completed exercises and so on. The states are updated when relevant changes occur. The database is situated on the web server. This way the user can continue his/her exercises on any computer with Internet access, whether he/she is at home, school or Internet cafe.

Accidents database contains statistics from real-life accidents. This information is used when evaluating the consequences of an in-game accident. The content of this database is not updated by users or their actions.

Graphics and sounds are stored in suitable file format and downloaded with the IRT e-Coaching package.

Environment Data contains the information about the milieu of the exercises. Basically the milieu is composed of different settings: urban, suburban, rural and woods. These parts are integrated in a natural manner into one fictitious town and its surroundings. The Environment Data contains the map with roads, locations of buildings, street signs etc. and the angles of inclination. Most exercises are limited to a restricted area of this map.

The Rule Base is a collection of event-trigger pairs. An event is composed of one or more conditions. When these conditions are fulfilled, the trigger goes off and causes one or more actions. The Rule Base is downloaded into the user's computer, and it cannot be updated by the users.

The IRT e-Coaching programme will in fact contain many separated rule bases. The rules that define the behaviour of pedestrians, other vehicles, traffic lights and so on have their own rule bases. So do the rules that define the consequences of the crash (the crash model).

The exercises are defined in a file, which is read by the Exercise Manager. The files contain information such as the locations of objects, the rules applied in this exercise as well as the graphics and sounds used. The exercise files are created and edited with the Authoring Tools by creating different combinations of predefined resources, such as objects (car with specific speed and direction), environmental variables (rain, oil spill on the road), locations (the exercise begins at specific part of the town) and rules (behaviour of other vehicles, the particular aspects of performance measuring in this exercise). Regular users however have no access to the Authoring Tools and cannot edit the contents of the exercise files.

7.4 Fulfilling the Requirements

The table 7.1 summarises which component of the proposed system architecture fulfills each requirement set in chapter 5.

Requirement	Responsible component
A1	e-Coaching, Exercise and Communications Managers
A2	Exercise Files
A3	e-Coaching, Exercise and Communications Managers
A4	Exercise Manager and Instructor
A5	e-Coaching, Exercise and Communications Managers
A6	e-Coaching, Exercise and Communications Managers
A7	Data storages, like exercise files and rule base
A8	User Inputs Handler, e-Coaching Manager and Visual Output
B1 – B4	User Inputs Handler and e-Coaching Manager
B5	The dynamic model of the motorcycle
B6 – B7	User Inputs Handler, e-Coaching Manager and Visual Output
C1	Rule Base
C2 – C3	Performance Evaluator
C4	Instructor
C5	Rule Base
C6	Performance Evaluator and Rule Base
C7	Performance Evaluator, Rule Base and e-Coaching Manager
C8	Safety Bubble
C9	Accidents database
C10	Crash Model
C11	Performance Evaluator
C12	The dynamic model of the motorcycle
C13	The dynamic model of the motorcycle and Visual Output
D1	e-Coaching Manager, Performance Evaluator and Visual Output
D2	e-Coaching Manager and User Inputs Handler
E1 – E4	Visual Output
E5	Data storages and Visual Output.
E6	Environment Data and Visual Output
E7	Visual Output and User Input Handler
F1 – F2	Visual Output
G1	Audio Output
H1 – H2	User database
I1 – I3	Website and Telecommunications
I4	Website

Table 7.1: Fulfilling the requirements of the IRT e-Coaching programme.

7.5 Process Description

An illustration of the process description can be found in the appendix D. The numbers listed below correspond to the numbers found in the illustration. Please note that while an arch may contain many numbers, the actions related to them do not happen simultaneously, but one at a time in a numerical order.

The process description starts at a state where the program package has been downloaded to the user's computer and launched.

1. The Exercise Manager retrieves a list of the available exercises from the data storage.
2. The list is relayed via the e-Coaching and Communications Managers to the Visual Output, which displays the list on a monitor to the user.
3. Using the mouse and/or the keyboard, the user selects an exercise. The User Inputs Handler interprets this request and sends it through Communication Manager to e-Coaching Manager, which requests the Exercise Manager to parse the selected exercise.
4. The Exercise Manager retrieves the definition file of the selected exercise from the data storage.
5. The Exercise Manager reads the exercise definition file and gathers the resources required by this exercise from the data storage and models. From these resources it parses the exercise package.
6. The Exercise Manager sends the exercise package to the e-Coaching Manager and asks it to run the exercise.
7. Based on the exercise package, the e-Coaching Manager creates a runtime database, Internal Object Data Model, into which it stores the states of all the objects used in this exercise. These objects include, for example, the motorcycle, other vehicles and pedestrians.
8. Via the Communications Manager, the e-Coaching Manager asks the Graphic and Audio outputs to produce the images and sounds for the user. The outputs are based on the states of the objects stored on the IODM.
9. The user reacts to the images on the monitor and starts to control the motorcycle by pressing keys on the keyboard. The User Input Handlers interprets these commands and sends them through the Communications Manager to the e-Coaching Manager.

10. When the user's inputs reach the e-Coaching Manager, it asks the relevant mathematical models to calculate the effects of these inputs.

11. Based on the effects of the inputs, the e-Coaching Manager updates the states of the objects in the IODM. If it later needs to know the state of one of the objects, it can query this information from the database.

12. The e-Coaching Manager sends relevant information to the Performance Evaluator, which uses the information to evaluate how safely the user is riding.

From this point on the system enters a loop, which repeats the phases 8-12 until the exercise ends. The exercise ends when the Performance Evaluator detects that the user has either completed or failed the exercise.

13. The Performance Evaluator notifies the e-Coaching Manager, that the exercise is finished.

14. The e-Coaching Manager erases the IODM, since it is no longer needed.

15. The Performance Evaluator creates a roundup of the user's performance and sends it to the Instructor.

16. Based on the roundup, the Instructor selects a fitting set of feedback from a predefined list of feedbacks, and sends it to the e-Coaching Manager, which relays the feedback to the Visual Output, which displays it for the user.

17. The e-Coaching Manager sends the results through the Communication Manager and Telecommunications to the server, where the data is used to update the user’s state (mark the exercise completed) and to show the score on the top score list.

7.6 Implementation Technologies

The suggested design can be implemented with a variety of different technologies. Each of the technologies has its own traits, making it difficult to judge which one would ultimately be the best one. The best implementation technology might simply be the one that the implementation crew is most familiar with, as knowing the knacks of specific technology may have a huge impact on the quality of the product.

Java™ is a programming language by Sun Microsystems™. Java is well suited for web usage, as the Java applications are first compiled to bytecode, which is later compiled to machine code by OS dependent virtual machines. This process makes the applications in principle OS independent, although this independency is easily obstructed by the use of additional OS specific Application Programming Interfaces (API). Java applications can be embedded to web page (applet) or run on a separate window (using Java Web Start technology).

Java compiler and virtual machines are provided under GNU General Public License (GNU GPL) by Sun Microsystems. As a full-scale programming language, Java is capable of solving complicated problems. Java's standard library offers ready-made tools for many routine tasks, like input handling. Many other APIs are provided by a devoted fan base. On the other hand to run Java applications, installed virtual machine is required.

Adobe® Flash® is a multimedia authoring system, supporting vector and raster graphics, streaming audio and video and its own script language, ActionScript. Like Java, Flash uses a virtual machine called Flash Player to run the applications. Flash Player is available as a free plug-in for the most common web browsers.

The strengths of Flash lie in its small file size, quick load times and lightweight resource usage. However, as Flash was initially created to display animations, it offers limited capacity for interaction and programmability. It also scales poorly on larger projects. In addition, creating Flash files requires commercial software.

Adobe® Shockwave® is another multimedia player, similar to Flash. Shockwave was originally intentioned for creating multimedia content to be distributed on CDs. It is therefore more versatile, making creation of more complex applications possible, although not in the same scale as Java.

Unlike Flash, Shockwave's rendering engine is capable of hardware-accelerated 3D-graphics. All the advantages the Shockwave has compared to Flash come with a price, particularly larger file size and longer loading times. The Shockwave's development software is also more expensive than that of Flash. Finally, the required plug-in, Shockwave Player, isn't as commonly in use as Flash Player, and isn't available for Linux OS at all.

Adobe Flex is a collection of different technologies that allows Web application developers to quickly and easily build rich Internet applications, which are deployed as Flash files. The newest version, 2.0, has greatly improved the product by introducing a new Eclipse-based IDE and a new version of ActionScript. Based on a preliminary study, Flex seems to be designed primarily for developing 2D datadriven applications. It is likely to be unsuitable for creating a complex system containing 3D-graphics and dynamical modeling. If further advances in the development of Flex take place, it is recommended to review its suitability for the IRT Project.

7.6.1 Technological Framework

In an ideal situation there would be a standardised web-based technological framework, which would conjoin the different implementation technologies and specify the rules for their interoperation. Extensible Modeling and Simulation Framework (XMSF) is a promising candidate for such framework.

XMSF provides the technical basis for transformational interoperability via XML interchange, profiles, and recommended practices for web-based modeling and simulation. Broad technical interoperability is provided by open standards, XML-based markup languages, Internet technologies, and cross-platform Web services. XMSF supports diverse distributed modeling and simulation applications. It also enables simulations to interact directly and scale appropriately over a distributed network through composable and reusable model components. In addition to employing mainstream practices of enterprise-wide software development, XMSF provides support for all types and domains of modeling and simulation. (Buss & Ruck, 2004.) However, the suitability of XMSF for the IRT Project has to be studied further.

7.6.2 Application Logic

The e-Coaching Manager, Communications Manager and Exercise Manager are relatively complex, and therefore Java is likely to be superior implementation technology. Java is a full scale programming language, whereas Flash's ActionScript and Shockwave's Lingo are simpler scripting languages. If these parts are implemented with Java, it is natural to create the rest of the application with Java as well.

7.6.3 Communications

User Inputs Handler

Java Foundation Classes (JFC) includes the Abstract Windowing Toolkit (AWT), which in turn includes an interface for input devices like keyboards and mice. Java uses event-driven architecture. When the user moves the mouse or presses a key on the keyboard, ATW creates an event. The nature of the event is then determined by Events

Dispatching Thread (EDT) and appropriate actions taken. This functionality is named User Inputs Handler in this document.

Visual Output

The visualization of the IRT e-Coaching programme can be implemented in many ways. Most of these ways are based on a high-level API, that uses lower-level APIs, namely Open Graphic Library (OpenGL) and Direct3D (D3D). OpenGL is a standard specification defining a cross-language cross-platform API for writing applications that produce 3D computer graphics (and 2D graphics as well) (Wikipedia, 2006). D3D is a part of DirectX (DX), which is API provided by Microsoft for handling tasks related to multimedia. Thus the visualisation can be seen as a layered model, illustrated in figure 7.2.

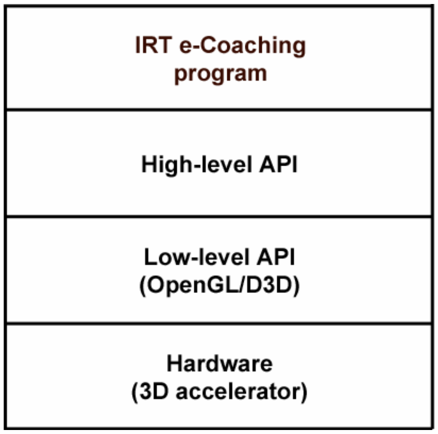


Figure 7.2: The layered model of the visualisation.

Few things need to be considered while deciding whether to use OpenGL or D3D. D3D is available only for Microsoft Windows Operating Systems, where as OpenGL has implementations for Windows, Linux, MacOS and other operating systems. OpenGL is more suitable for projects where the hardware configurations of the end users vary, because OpenGL ensures that the required features can be produced using software emulation if no direct hardware support is available. Selecting OpenGL also provides wider selection of high level APIs for Java environment.

Java 3D is a 3D API developed under the Java Community Process. JOGL is a low-level wrapper API developed by Sun Microsystems. Espresso3D is a high performance real-time 3D engine developed by Curt Moxley. Many other such APIs exist. In the end this means that pinpointing a single best option is difficult, and the choice should be made based on which technology the implementer is accustomed to.

Audio Output

Java provides a sound extension for playback of audio files, such as Waveform Audio Format (WAV). This extension isn't included in the JRE, and thus increases the size of the downloaded file. Even then this is more convenient approach than programming an own audio output handler.

Telecommunications

In the context of the IRT e-Coaching programme, telecommunications means two separate things: Firstly it means the interaction the user has with other users via the website of the programme. As this interaction is done through the web browser, it is not of interest here.

Secondly telecommunications means the messages that the launched the IRT e-Coaching programme software exchanges over the Internet with the server in order to enable the functioning of the software. Once again many different technologies are available for this purpose. One such technology is Simple Object Access Protocol (SOAP). SOAP is a protocol for exchanging XML-based messages over computer network, normally using HTTP (Wikipedia, 2006c). SOAP is simple, flexible and extensible. As it is based on XML, it is programming language, OS and hardware independent. SOAP with Attachments API for Java (SAAJ) provides an easy way to write SOAP messaging applications with Java.

7.6.4 Databases

Internal Object Data Model

As relational databases are the de facto standard for database architecture today, they are an obvious choice. Management of the relational databases of the IRT e-Coaching programme can be implemented with such Relational Database Management Systems as MySQL or PostgreSQL, as their usage is free under GNU GPL and Berkeley Software Distribution (BSD) license, respectively.

Other Databases

Like the IODM, the user data and accidents databases should be implemented with MySQL or PostgreSQL. The User Data is a permanent database located on the web server. The location of the accidents database isn't that obvious. It can be located on the web server, but this might cause a small delay when processing the results of an accident. On the other hand, the database can be downloaded on the user's computer, but this will build up the size of the downloaded file.

Database Communications

Java includes a Java Database Connectivity (JDBC) API, which offers simple methods for accessing a relational database. This can be used to control the IODM database as well as the data storages.

7.6.5 Data Storages

Exercise Files

The exercise files should be written in a format that can be quickly read by the Exercise Manager, easily edited by the Authoring Tools and stored in a relatively small space. The file format must not restrict the things that can be said about the exercises (e.g. failure conditions, objects involved). Extensible Markup Language (XML) fulfills these requirements. XML is a general-purpose markup language recommended by W3C. Depending on the size of the exercise files, they can either be read from the server only when needed or be downloaded to user's computer as one package when the programme starts.

Graphic Files

The graphics, which include 3D models and 2D images, are stored in files. The 3D model files include the wireframe model of the objects (the motorcycle, cars, houses and so forth) and the textures that cover these models, creating the appearance of the object.

The 3D models can be created using various technologies and applications. In polygonal modeling the models are created by approximating their geometry with polygons. Polygonal modeling is the most common modeling method. It is easy to use, but the models tend to be angular. Non Uniform Rational Basis Plane (NURBS) is another method, which uses a set of curves to create the models. NURBS is a direct opposite of the polygonal modeling: creating of the models tends to take more time, but the models aren't angular. The models are also more complex and therefore more calculating power is needed when rendering the image. The models can also be created by reverse engineering, where a real-life object is digitalised with laser or optical scanner. This however requires special equipment and the created models usually need to be finished manually with modeling software. As the graphical splendour is not the focus of the

IRT e-Coaching project, the polygonal modeling is recommended, since it creates simple, light-weight models.

As stated before, there is a multitude of different software for creating 3D models. The most popular commercial 3D software is probably 3D Studio Max® by Autodesk®. Once again the best software depends on the preferences of the implementer, since the method used for creating the models is not as important as the quality of the models created.

It is also noteworthy that the models can be bought. Since the objects in the programme are quite common ones in video games, availability shouldn't be a problem. Buying the models instead of creating them from scratch can save time and money and even result in better quality models.

Some 2D graphic files are required as well, for example for the backgrounds. The relevant thing about these files is their format, as this might affect the quality of the image. Different formats are designed for different purposes, and therefore the format has to be chosen in each case separately.

Audio Files

The sound environment of the programme is rather minimalistic, consisting mainly of the sound of the motor. These few audio files can be stored in uncompressed WAV-format. If the size of the audio files grows high, a compressed audio format has to be considered. This however increases the processing at the user's end, as the sound files have to be decompressed.

7.6.6 Models

Dynamic Models

After the mathematical modeling of the motorcycle and other objects is done, the models can be designed and tested using Mathworks' Matlab and Simulink software. The models are converted into embedded packages, which can then be used without Matlab or Simulink. Simulink can create these packages automatically as C code. If a Java implementation is needed, it must be done manually.

The dynamic model packets must be downloaded to the user's computer, since a delay caused by Internet connection is unacceptable when a response by the model is needed. The complexity of the model increases the size of the package and the calculation power needed to use it. Therefore a compromise between the complexity of the model and the authenticity of the response has to be made.

Rule Base

The Rule Base is implemented as a reactive rule engine, i.e. it detects and reacts to interesting events. Rule engine separates the rules themselves from the application code, thus making it easier to edit the rules, since no programming is required to do so.

There are many rule engines available on the market. JBoss Rules is an ideal solution for the IRT e-Coaching project, because of the open source Java implementation. JBoss Rules uses the Apache Software Foundation open source license, thus making it free to use. Java implementation ensures smooth interaction with the other Java components of the programme.

Safety Bubble Model and Crash Model

The Safety Bubble and Crash Models are rule bases specialised in their respective fields. They are created only to fulfill the responsibilities defined in subsection 7.2.4. and no other tasks.

7.6.7 Authoring Tools

If the exercise files are written in a plain XML format, they can basically be edited by a regular text editor. Some free text editors offer an XML-highlighting, thus making it easier to read and edit the XML files. However, if the exercise files are created by technically less experienced user or if the files are edited so frequently that using a regular text editor is too tiresome, special Authoring Tool might be valuable. Since the Authoring Tool would be a separated application, unconnected to the rest of the programme, the implementation technology can be chosen freely.

7.7 Conclusion

The web-based implementation seems auspicious for the IRT e-Coaching project. Rapid changes, constant advances, developing technologies and new standards are characteristic for the web, and therefore the optimal implementation technology shouldn’t be decided until the project is actually launched. The suitability of the technologies recommended here must be tested with prototypes before the final decision.

8. COST ESTIMATE

Estimating the amount of work involved in software project is always a challenging task. Especially at the earliest stages of the development, uncertainty causes great vagueness in the estimations. At this point, estimates with an accuracy of plus or minus 50 percent are typical for an experienced estimator using informal techniques such as historical comparisons or group consensus. Therefore these estimates must be reviewed at later stages of the project in order to take corrective action in time. (Roetzheim, 2001.)

Since the accuracy of the estimates is imprecise at this stage, the amount of work required to complete the software is estimated in man-months. When the time required to complete each component of the system is estimated separately (see table 8.1), a total amount of work sums up to 21 man-years. As said before this estimate is extremely crude. Many factors, such as the implementation technology, the experience of the implementer and unforeseen complications contribute to this uncertainty. Using premade components, such as the 3d models of the vehicles, can save time and money and should be taken advantage of when ever possible.

Component	**Man-months:**
e-Coaching Manager	21
Internal Object Data Model	15
Performance Evaluator	12
Instructor	6
Communications Manager	9
User Inputs Handler	6
Visual Output	18
Audio Output	6
Telecommunications	12
Exercise Manager	9
Exercise Files	12
Authoring Tools	9
Rule Base	24

Environment Data	12
Graphic Files	30
Audio Files	3
Accidents Database	12
User Data	3
Dynamical Models	15
Safety Bubble Model	9
Crash Model	9
Total:	**252**

Table 8.1: Time required to complete the components.

The cost of one man-year can be estimated to be about 72,000€ (6,000€ per man-month). On an EU project this includes only the real expenses. The best case estimate (work estimate has 50% excess) is then 10.5 man-years * 72,000 €/year = 756,000€. Similarly the expected case estimate (work estimate is accurate) is 21 man-years * 72,000 €/year = 1,512,000€. The worst case estimate (workload has 50% deficit) is then 31.5 man-years * 72,000€/year = 2,268,000€. Thus the expected cost of the labour is about one and a half million euros.

The labour is the single largest contributor to the cost of the IRT project, but other expenses exist as well. Depending on the choice of implementation technologies, some software applications used in the development are quite expensive. These commercial solutions should be deployed only when no free counterpart fulfils the requirements of the project.

The deployment of the servers that host the website and run the IRT e-Coaching programme naturally causes more expenses. They also require ongoing upkeep.

9. OPPORTUNITIES AND CHALLENGES

The following opportunities and challenges have been identified while producing this evaluation.

9.1 Opportunities

- Consumer priced hardware develops rapidly. More calculation power and faster Internet connections can be acquired with a lower budget. This makes it easier to reach the target group with simulator software.
- Advances in standardisation makes it possible to utilise compatible premade modules in the future.
- The development of the new generation of video game consoles opens interesting possibilities to take advantage of the user friendly package of state of the art graphics, premade game engines and communal collaboration aspects.

9.2 Challenges

- In order to create realistic, immerse learning environment, the number of required training settings and objects may rise to very demanding levels.
- Deriving the rules from the dynamical models of the motorcycle and other vehicles is a challenging, yet critical task.
- Creating the artificial intelligent behind the rules of hazardous situations is laborious assignment.
- Fitting the high-level dynamical models, 3d models, real-time simulation and database utilisation into a web-based application is challenging.
- Some features are limited by the sandbox paradigm of the Java applications.
- A detailed synopsis and requirements specification are required to calculate a precise cost estimate.
- Since the commercial companies don't share the details of their products, the amount of work required to produce the IRT e-Coaching programme is hard to estimate.

10. SUMMARY

Currently initial riding trainees in different member states of the European Union have unequal opportunities for good, affordable and comprehensive training. Existing educational methods have difficulties reaching the large homogenous target group. The IRT e-Coaching programme working on the users' existing PCs is an attractive solution for offering equal access to training in hazard perception and correct attitude and behaviour in traffic. The IRT e-Coaching programme can be seen as particularly attractive training method for young trainees of age 15-35.

The IRT e-Coaching programme is ideal for training hazard perception and avoidance and correct attitude and behaviour in traffic. With a programme the trainee can be exposed to hazardous situations without real danger to the trainee or other road users. Situations that would rarely occur in real life can be easily produced and replayed until the trainee can handle the situation safely. Observing the performance of the trainee and giving feedback is easier and more illustrative within the programme than it would be in real life. These factors would make The IRT e-Coaching programme an attractive addition to existing training methods.

We studied the PCs and video game consoles as platforms, as well as the possibilities of internet-based solutions. PC hardware seems more suitable for the IRT e-Coaching programme than the video game consoles due to wider availability and adequate calculation power. They also provide an easy access to the Internet, which can be used to create communal collaboration among the users of the programme, thus reinforcing the learning process. Wide range of open source components support the selection of the PCs. The royalties to the manufacturers of the consoles and non-existent interoperability of the consoles hinders their suitability as the platform of the IRT e-Coaching programme.

It is recommended that the programme is distributed via a website since this is an ideal way to distribute the software to the initial rider trainees all around Europe. The website also functions as a base of operations for the communal collaboration aspects. the IRT e-Coaching programme itself consist of two ”modes”: In a level-based mode the trainee completes a series of levels with randomly generated traffic situations with increasing difficulty. In the exercise-based mode the user can select an interesting aspect of riding, and the exercise then contains traffic situations related to this aspect. Before and after every level or exercise a briefing and debriefing is held.

Sun Microsystem's Java is assessed to be the most suitable implementation technology due to its suitability for web development, cross-platform support, capability to cope with complex projects and the wide range of premade APIs and integrated development environments available. The creation of dynamic models is most efficient with Mathwork's Matlab and Simulink software. JBoss Rules is a appropriate solution for creating the required rule bases. The interoperability of this setup has to be tested before the implementation process can be fully started.

The cost estimate for this project, executed as recommended in this document, is 1,512,000€. Since the estimation of workload at the earliest stages of software project is extremely difficult, a 50% safety margin is applied for this estimation.

The process of this evaluation leads to a recommendation. It is based solely on theory and previous research of the topic. Therefore it is crucial to create prototypes and technical tests before the final implementation of the the IRT e-Coaching programme is undertaken. More detailed requirements of the IRT e-Coaching programme are required in order to accurately evaluate the suitability of the technical choices. We believe that this evaluation will aid in further specifying these requirements.

APPENDIX A: DEFINITIONS, ACRONYMS AND ABBREVIATIONS

ACEM	Association des Constructeurs Européen des Motorcycles, the European motorcycle manufacturers' organisation.
API	Application Programming Interface, a documented interface which provides access to functions without requiring an understanding of the functions internal implementation.
ATW	Abstract Windowing Toolkit, Java's original platform-independent windowing, graphics, and user-interface widget toolkit.
BSD	Berkeley Software Distribution, a Unix derivative distributed by the University of California, Berkeley.
CPU	Central Processing Unit.
DX	DirectX, collection of multimedia related APIs for Microsoft Windows.
D3D	Direct 3D, a graphics API, which is a part of DX.
EDT	Event Dispatching Thread, a background thread used in Java to process events from the event queue.
Exercise-based mode	The part of the IRT e-Coaching programme where the user can select which specific topic he/she wants to practise. The available topics are listed on requirement A2, see subsection 5.1.1.
FEMA	Federation of European Motorcyclists' Associations.
FIM	Fédération Internationale des Motocyclisme.
GNU	GNU's Not Unix, a free operating system. See http://www.gnu.org/ .
GPL	General Public License, see http://www.gnu.org/licenses/gpl.html .
GPU	Graphics Processing Unit.
IDE	Integrated Development Environment, a combination of tools that form an entity for developing software.
IODM	Internal Object Data Model, see section 9.1.1.
IRT	Initial Rider Training -project
IVV	The International Association for Driver Education
Iteration	In software development, iterative is used to describe process where an application is developed in small

	sections called iterations. Each iteration is reviewed and critiqued; insights gained from the critique of an iteration are used to determine the next step in development.
JFC	Java Foundation Classes, a graphical framework for building portable Java-based graphical user interfaces.
JRE	Java Runtime Environment, software package required to run Java applications.
Level-based mode	The part of the IRT e-Coaching programme where the user tries to complete a series of randomly generated levels.
Mentor	Mentor is the embodiment of the instructor in the IRT e-Coaching programme. A renowned professional motorcycle rider like Valentino Rossi, who gives instructions and feedback to the user.
OECD	Organization for Economic Cooperation and Development.
OpenGL	Open Graphics Library, a cross-platform 3D graphics API.
OS	Operating System.
SAAJ	SOAP with Attachments API for Java, an API for writing SOAP messaging applications.
SOAP	Simple Object Access Protocol, a protocol for exhanging XML-messages over computer network.
RAM	Random Access Memory.
TUT	Tampere University of Technology.
UI	User interface.
Vägverket	The Swedish Road Administration
WAV	Waveform Audio Format
WWW	World Wide Web.
W3C	World Wide Web Consortium, main international standards organization for the WWW.
XML	Extensible Markup Language, general-purpose markup-language recommended by W3C.
XMSF	Extensible Modeling and Simulation Framework.

APPENDIX B: REFERENCES

Bareiss, R. & Osgood R. 1993. Applying AI Models to the Design of Exploratory Hypermedia Systems. Proceedings of Hypertext '93, Seattle, Washington, US, November 14-18.

Buss, A. & Ruck, J. 2004. Joint Modeling and Analysis Using XMSF Web Services. Proceedings of the 2004 Winter Simulation Conference.

CIA, Computer Industry Almanac. 2006a. PCs In-Use Surpassed 900M in 2005. Available as a WWW resource: [http://www.c-i-a.com/pr0506.htm]. Cited Dec 4th, 2006.

CIA, Computer Industry Almanac. 2006b. 1.3B Cumulative PC Sales Over Next 5 Years. Available as a WWW resource: [http://www.c-i-a.com/pr0606.htm]. Cited Dec 4th, 2006.

Cormier & JD Hagman. 1987. Transfer of Learning.

de Corte, E. 2003. Transfer as the productive use of acquired knowledge, skills, and motivations. Current Directions in Psychological Science Volume 12 Issue 4 Page 142.

Grabinger, R.S. & Dunlap, J.C. 1995. Rich environments for active learning: a definition, Association for learning a Technology journal 2.

Harper, B., Squires, D. & McDougall, A. 2000. Constructivist simulations in the multimedia age. Journal of Educational Multimedia and Hypermedia, Association for the Advancement of Computing in Education, 9, pp. 115-130.

Hatakka, M., Keskinen, E., Gregersen, N.P., Glad, A., Hernetkoski K. 1999. Results of EU-project GADGET, Work Package 3.

Hoeschen, A. & Bekiaris E. 2001. System for Driver Training and Assessment Using Interactive Evaluation Tools and Reliable Methodologies. TRAINER deliverable D2.1. Institut fuer Arbeitsorganisation and der Universitaet Dortmund, Dortmund, Germany.

Huhtamäki, J., Ranta, P. & Pohjolainen, S. 2005. On the modeling-based framework for the instructional assessment of dynamic processes. The 3rd International Conference on Multimedia and

Information & Communication Technologies in Education (m-ICTE2005) Cáceres, Spain on June 7-10th, 2005.

Jain, Sanjay & McLean, Charles R. 2004. An Architecture for Integrated Modeling and Simulation for Emergency Response. Proceedings of the 2004 Industrial Engineering Research Conference May 15-19, Houston, Texas.

Jonassen, David. 1995. Computers as Cognitive Tools: Learning with Technology, Not from Technology. Journal of Computing in Higher Education, v6 n2 p40-73 Spring 1995. ISSN-1042-1726.

Järvinen, Pertti. 2000. Tutkimustyön Metodeista. Opinpajan Kirja, Tampere. ISBN: 951-97113-8-4.

Lehtinen E. 2000. Pedagogical guidelines for designing computer-based learning environment for complex learning. PEG 2001. eds. Ruokamo H. et al.

Microsoft Corporation. 2006. Xbox.com news. Available as a WWW resource: [http://www.xbox.com/zh-SG/community/news/2006/20060510.htm]. Cited Dec 12th, 2006.

Norwegian Public Roads Administration report. 2005. Norwegian Public Roads Administration, Directorate of Public Roads.

Nählinder, S. 2006. Training of fighter pilots in the Swedish Air force. SIMSEM 2006 in London on 26.-28.9.2006. Available as a WWW resource: [http://www.norsim.org]. Cited Jan 9th, 2006.

OECD, Organization for Economic Cooperation and Development. 2006. OECD Broadband Statistics to June 2006. Available as a WWW resource: [http://www.oecd.org/document/9/0,2340,en_2649_34225_37529673_1_1_1_1,00.html]. Cited Dec 4th, 2006.

Pohjolainen, S., Hautakangas, S., Ranta, P., Levasma J. & Pesonen. K. 2001. A Learning Experiment in Mathematics using A&O-Learning Environment. PEG 2001. eds. Ruokamo H. et al.

Pohjolainen, S., Suomela, K. & Häkkinen P. 2003. Mathematical Modelling over the Internet - a National Network Project of Finnish Virtual University. 11th International PEG Conference.

Prensky, Marc. 2001. Digital Game-Based Learning. McGraw-Hill Companies. ISBN: 0071363440.

Ranta, P. 2003. Possibilities to develop forest machine simulator based studying. Proceedings of PEG 2003. Powerful ICT Tools for Teaching and Learning

Ranta, P. 2005. Operators´ tacit knowledge supports forestry machine simulator based education. p. 39-48.in Ranta P. 2005. ed. Proceedings of the International Seminar on Simulator-Based Training of Forest Machine Operators. Tampere University of Technology. Digital Media Institute. Hypermedia Laboratory. Report 1/2005.

Reed, M.P. & Green, P.A. 1999. Comparison of driving performance on-road and in a low cost simulator using a concurrent telephone dialing task, Ergonomics, 42(8) 1015-1037.

Roetzheim, William. 2001. Estimating Software Costs. Cost Xpert Group Inc. Available as a WWW resource: [http://www.costxpert.com /resource_center/SDarticles/SDArticle1.pdf].

Rothengatter, T. 1997. Psychological aspects of road user behaviour. Applied Psychology: An International Review, 46, 3, 223-234.

Schön, D.A. 1983. The Reflective Practioner: How professionals think in action. Cambridge, MA: Basic Books.

Sony Computer Entertainment Inc. 2006. Cumulative Production Shipments of Hardware / PlayStation 2. Available as a WWW resource: [http://www.scei.co.jp/corporate/data/bizdataps2_e.html]. Cited Dec 12th, 2006.

Sweller, J. 1988. Cognitive load during problem solving: Effects on learning, Cognitive Science, 12, 257-285.

van Emmerik, M.L., van Rooij, J. 1999. Efficient Simulator Training: Beyond Fidelity, Proceedings of the 10th International Training & Education Conference.

van Emmerik, M.L. 2004. Beyond the Simulator - Instruction for High-performance Tasks. Print Partners Ipskamp. ISBN: 90-365-2068-1.

Vartiainen, M. 1985. Simulaatio työtaidon kehittäjänä. TKK.

Wikipedia, the free encyclopedia. 2006. OpenGL. Available as a WWW resource: [http://en.wikipedia.org/w/index.php?title=OpenGL &oldid=92006684]. Cited Dec 8th, 2006.

Wikipedia, the free encyclopedia. 2006b. Platform (computing). Available as a WWW resource: [http://en.wikipedia.org/w/index.php?title=Platform_%28computing%29&oldid=90372115]. Cited Dec 11th, 2006.

Wikipedia, the free encyclopedia. 2006c. SOAP. Available as a WWW resource: [http://en.wikipedia.org/w/index.php?title=SOAP&oldid=94269630]. Cited Dec 20th, 2006.

Wikipedia, the free encyclopedia. 2006d. Playstation 3. Available as a WWW resource: [http://en.wikipedia.org/w/index.php?title=PlayStation_3&oldid=95683793]. Cited Dec 22th, 2006.

Wikipedia, the free encyclopedia. 2006e. Educational Technology. Available as a WWW resource: [http://en.wikipedia.org/w/index.php?title=Educational_technology&oldid=94893360]. Cited Dec 22th, 2006.

Wikipedia, the free encyclopedia. 2007. Educational Game. Available as a WWW resource: [http://en.wikipedia.org/w/index.php?title=Educational_game&oldid=100482049]. Cited Jan 15th, 2007.

Wikipedia, the free encyclopedia. 2007b. Edutainment. Available as a WWW resource: [http://en.wikipedia.org/w/index.php?title=Edutainment&oldid=96782323]. Cited Jan 15th, 2007.

Wikipedia, the free encyclopedia. 2007c. Computer Simulation. Available as a WWW resource: [http://en.wikipedia.org/w/index.php?title=Computer_simulation&oldid=99348889]. Cited Jan 15th, 2007.

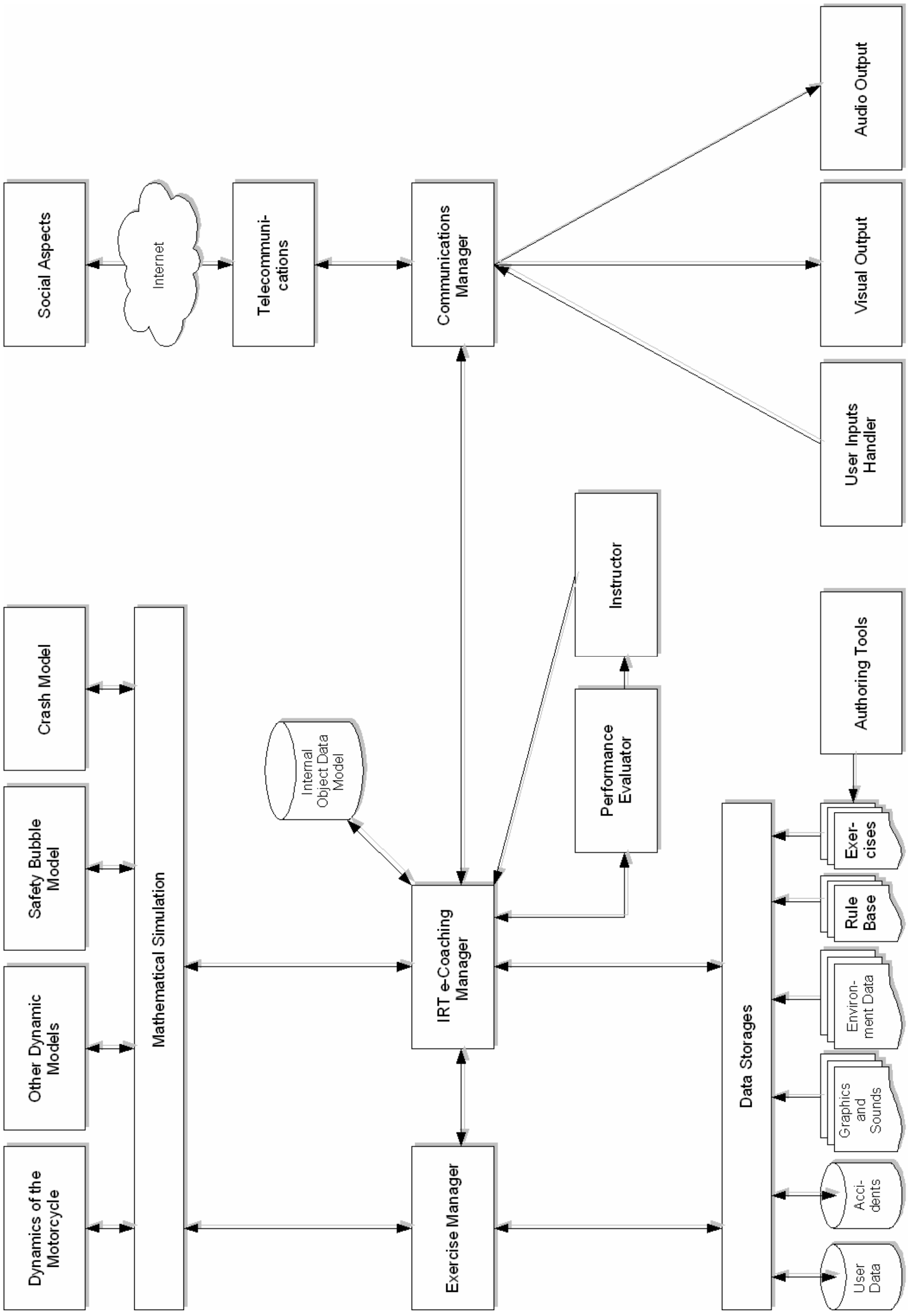
Wikipedia, the free encyclopedia. 2007d. Blended Learning. Available as a WWW resource: [http://en.wikipedia.org/w/index.php?title=Blended_learning&oldid=98878658]. Cited Jan 15th, 2007.

Woolf, B. P. 1988. Representing complex knowledge in an intelligent machine tutor, in: Self, J. (ed.) Artificial Intelligence and Human Learning. Chapman and Hall Ltd, London. pp. 3-27.

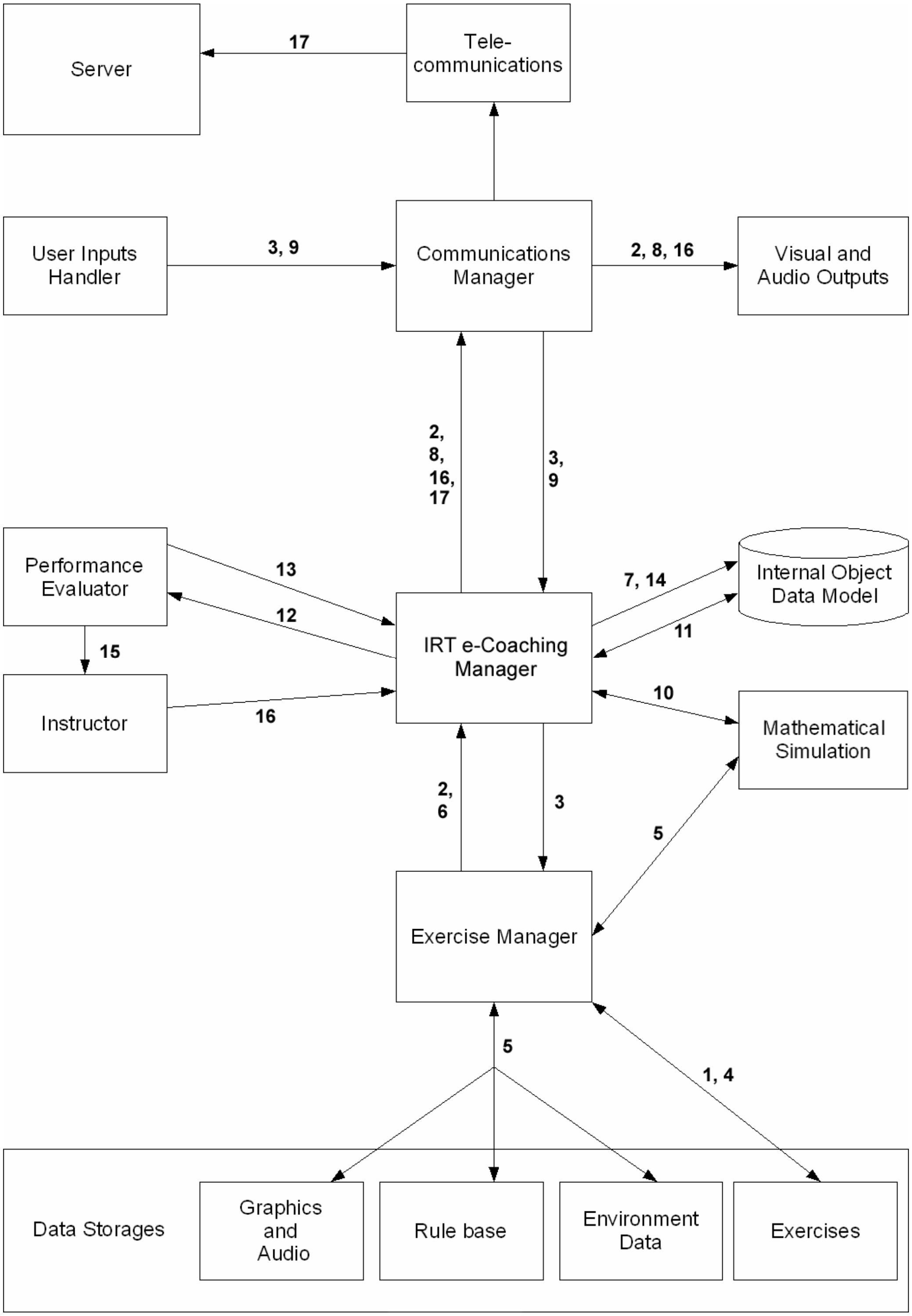
XNA FAQ. 2006. Microsoft Corporation. Available as a WWW resource: [http://msdn.microsoft.com/directx/xna/faq/]. Cited Dec 21st, 2006.

Youngblut, Christine. 1997. Educational Uses of Virtual Reality Technology. IDA Document Report Number D-2128. Available as a WWW resource: [http://www.hitl.washington.edu/scivw/youngblut-edvr/D2128.pdf].

APPENDIX C: ARCHITECTURAL DESIGN



APPENDIX D: PROCESS DESCRIPTION



APPENDIX E: SCENARIO SKETCHES

The following scenario sketches were used to create discussion in the meeting held on November.

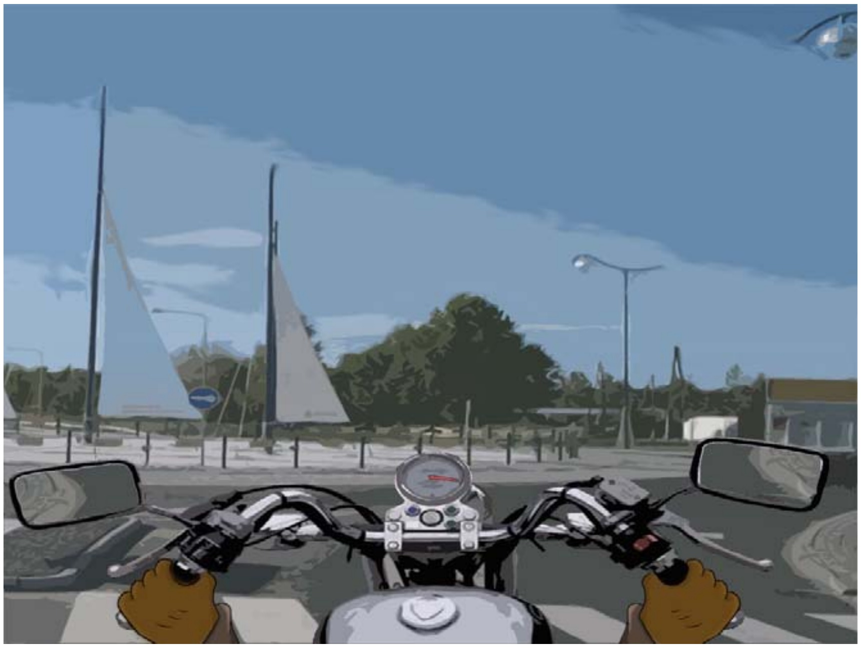


Figure E.1: Riding in a roundabout.

The roundabout scene can be used to teach proper situational speeds and percepting the surroundings. In order to know if it is safe to ride into the roundabout the user has to turn his head to the sides. In order to see what's going on behind, the user has to turn his head all the way back or express that he is paying attention to the rear view mirrors by clicking on them with the mouse.

Discussing the User Interface (UI) was easier with a concrete example. The required controls were identified and their implementation was discussed. To create a realistic riding sensation, the UI will be stripped-down of all unnecessary information, like a minimap or menus which can be found on regular motorcycle video games.

In the scene on figure E.2 the user is riding behind a bus. This scene contains multiple simultaneous hazards (the bus is about to stop, there's a kid on the side of the road and cars are approaching from the opposite direction). This can be used to teach hazard prioritising as well as that sometimes pulling over is the only safe action the rider can take.

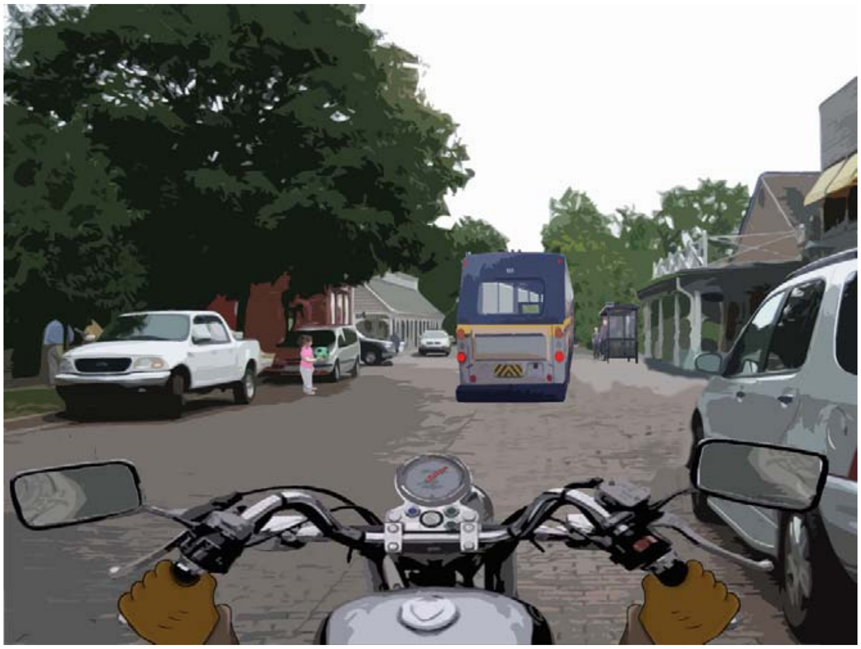


Figure E.2: Riding behind a bus.

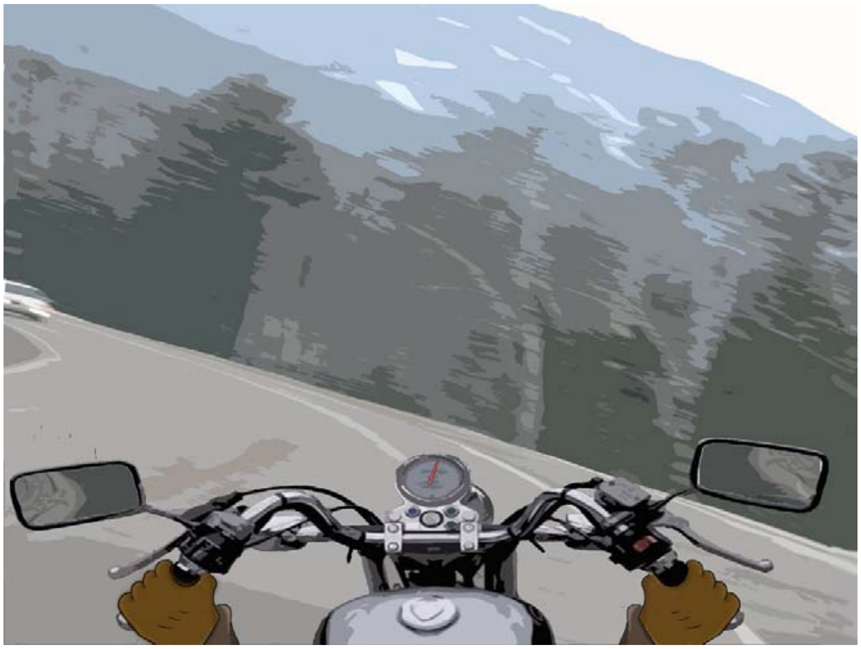


Figure E.3: Riding on a bend.

The figure E.3 demonstrates that in order to create a realistic sensation, the world gets tilted while the user is riding on a bend. The dynamic models are used to calculate the speed and other factors required for the user to lose control over the motorcycle. The three scenes above demonstrate the different surroundings the programme will contain: urban, suburban, rural and woods.