An evaluation of the potential of e-Coaching for Riders

a presentation to the
Directorate-General for Energy and Transport
July 17th, 2007
Brussels

Structure of the Presentation

1. Goals of the evaluation
2. Study Process and Methods
3. Requirements for IRT e-Coaching
4. Basic Functionality
5. Possible Implementation Technologies
6. Modular Architectural Design
7. Recommended Implementation Technologies
8. Evaluation Planning

1. Goals of the IRT e-Coaching evaluation

- Free simulation focused on teaching hazard perception and avoidance, as well as correct behaviour and attitude in traffic.
- Simulation which can be run on readily available personal computers or video game consoles.
- Learn through understanding the consequences of incorrect actions and provided feedback.
- Method for the trainees to experience authentic hazardous riding scenarios, which they are likely to face in real traffic.

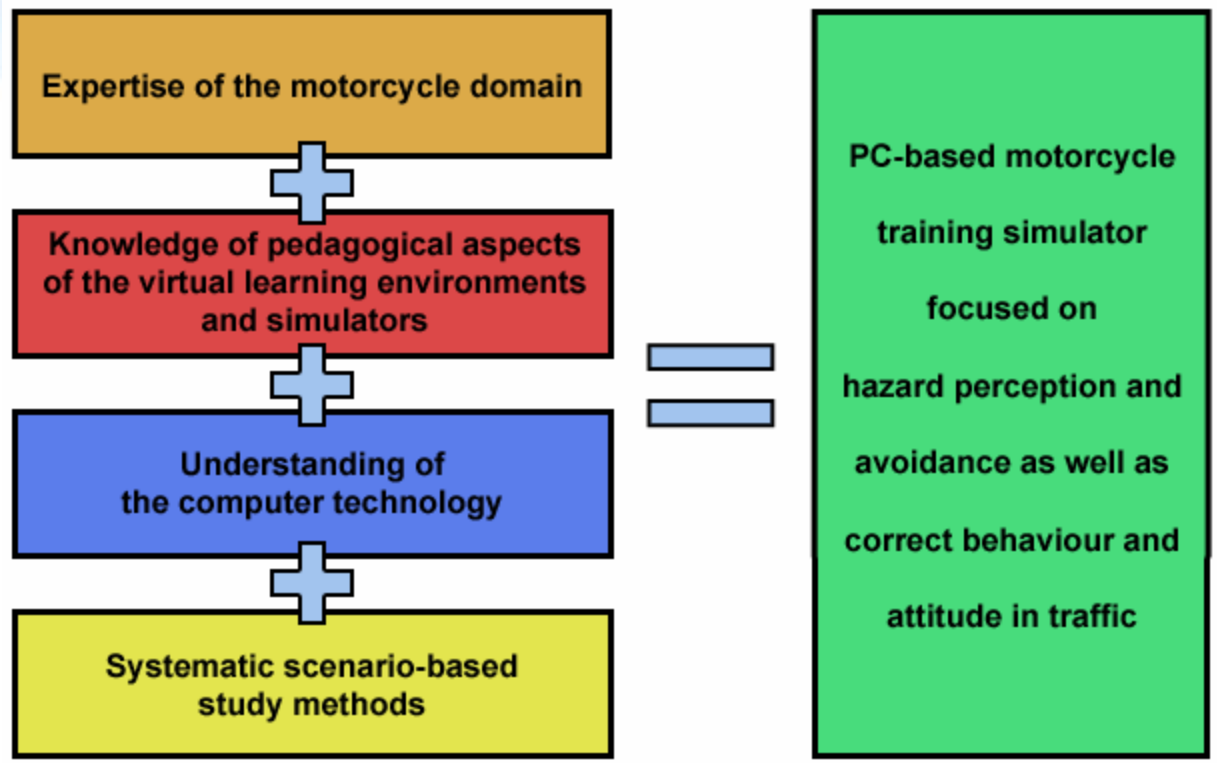
1. Goals of the IRT e-Coaching evaluation

- Enable an easy transference of the learned abilities to real-life situations.
- Targets mainly the young initial riders of the ”video game generation”.
- Be a self-learning process, rather than a replacement for the traditional instructor-based training.

What is the best way to achieve all this?

2. Evaluation Process and Methods

Interactive and iterative process composed of cycles of research, design and discussions.



2. Evaluation Process and Methods

Scenario-based discussions helped to clarify the focus of the project and all the factors that needed to be taken into account.

3. Requirements for IRT e-Coaching

To meet the goals, requirements for the simulation were set:

- Accessible to the wide audience.
- Self-learning process.
- Exercises created by competent instructors.
- The trainee can control the parameters of the environment.
- Evaluating the performance of the trainee.
- Giving feedback based on the performance.
- User interface, graphics and sounds.
- Communal collaboration.

What kind of functionality fulfils these requirements?

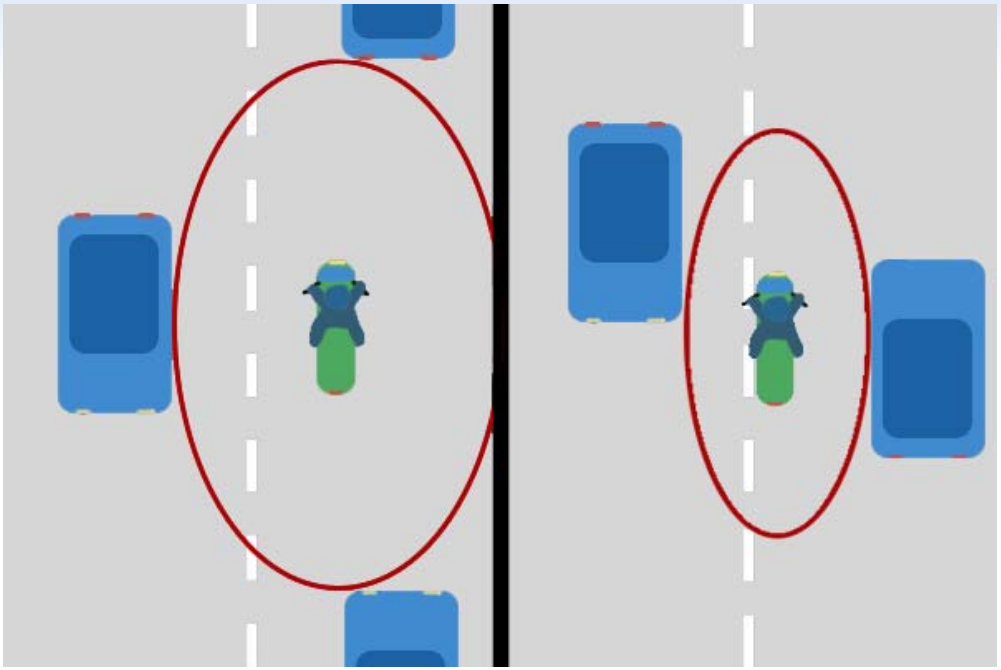
4. Basic Functionality

To meet the requirements, the following basic functionality was planned:

- The simulator is launched from a website.
- Two modes:
  - Level-based mode: practice random events with increasing difficulty
  - Exercise-based mode: practice specific manuscripted exercise
- Realistically behaving traffic
- Authentic riding environments.

4. Basic Functionality

- Riding and feedback evaluated based on the concept of the safety bubble.
- Constrain crash game behaviour.
- Simple controls suited for keyboards.



4. Basic Functionality

- Mentor offering help and tips.
- Minimal briefing and debriefing.
- Emphasize the consequences of the accidents: injuries and repair costs.
- Cartoon-like graphics with stripped-down user interface.

What is the best way to implement this functionality?

5. Possible Implementation Technologies

PC:

Pros:

- Vast numbers of PCs at homes, schools, libraries…
- People are used to computers
- Easy to connect to on the Internet
- Easily updated content

Cons:

- Clumsy controls
- Varying hardware & software configurations

5. Possible Implementation Technologies

Video Game Consoles:

Pros:

- Unified device base: impressive graphics etc.
- Versatile controllers
- Development toolkits might cut down the production time

Cons:

- Different consoles require different game versions
- Inferior numbers sold (when compared to PCs)
- Royalties
- Ties to project to console vendors

5. Possible Implementation Technologies

Internet-based approach:

Pros:

- Easy and economical distribution
- User data stored on server: practice anywhere you want
- Communal collaboration and social aspects
- Web standards: guaranteed interoperability in the future

Cons:

- Slow connections may cause problems
- Requires servers

5. Possible Implementation Technologies

Recommendation:

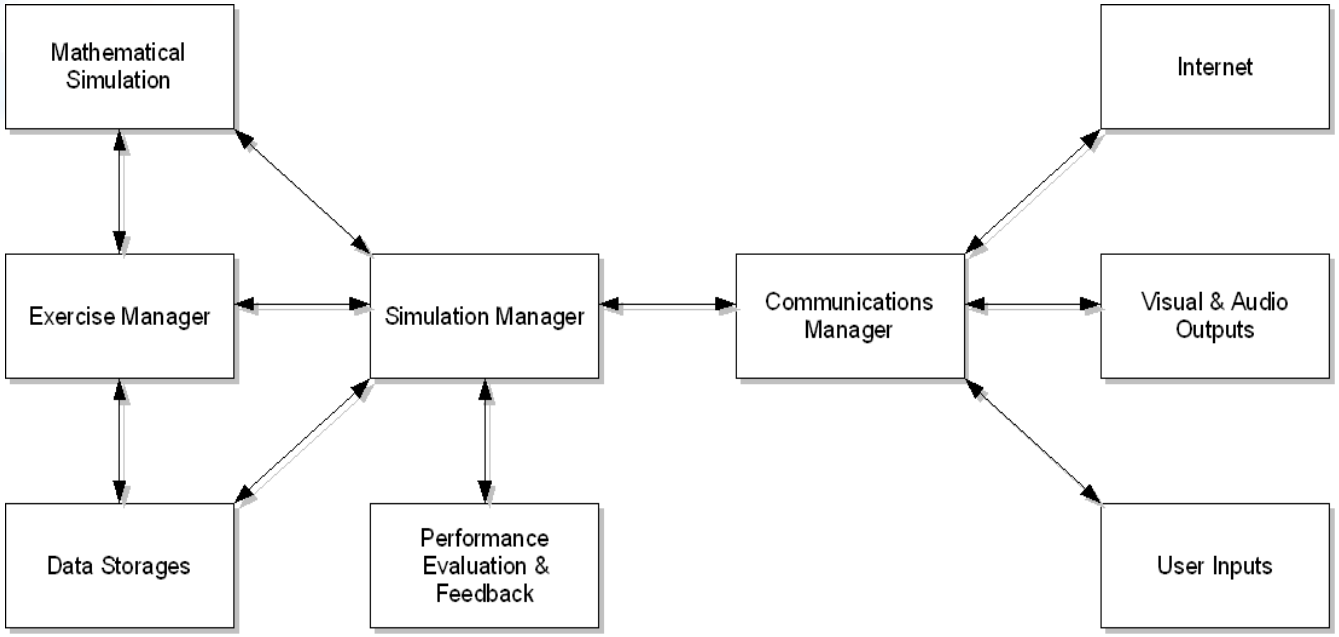
- To easily reach the young initial rider trainees around Europe
- To fully utilise the Internet and communal collaboration
- To easily update the simulation and exercises in the future
- To be independent of any single vendor and their decisions

Internet-based PC solution.

6. Modular Architectural Design

Modular Architectural Design:

Dividing the program into independent modules with specific responsibilities.



6. Modular Architectural Design

Recommended because:

- Makes it possible to create and test the modules separately.
- Separates the content and the program logic: creating new exercises and editing existing ones is possible.
- Software is easier to update and maintain.

7. Recommended Implementation Technologies

Modular architecture implemented with Internet-based technologies:

- Operating system independent programming languages and tools
- Open sources & free licenses
- Standards minimize the future problems
- Website for communal collaboration
- Databases for user data and statistics

For more details, please see the IRT e-Coaching evaluation report

8. Developing IRT e-Coaching

- Challenging project.
- Implementation requires an interdisciplinary workgourp.
  - Defining the exercises, curriculum and other education aspects
  - Modeling traffic events, vehicles etc.
  - Technical know-how
  - Graphic design
  - And more

8. Developing IRT e-Coaching

Cost estimate

- The costs are extremely difficult to estimate on the early stages of a software project
- Safety margins of ±50%
- Revisions at the later stages

- 21 man-years
- 72.000€ per year
- Around 1.500.000€ total (±750.000€)

Schedule

- Large scale project – more planning required
- Probably a 3 year schedule

Tampere University of Technology support the IRT Project's Deliverable 4 recommendation and would welcome an ongoing involvement

Thank You!

Senior Researcher
Pekka Ranta

TUT/Hypermedia Laboratory
PL 553
33101 Tampere
Finland

phone +358 3 3115 5606
mobile 040-849 0173
email pekka.a.ranta@tut.fi