USER REQUIREMENTS TO MODEL SCENARIOS ON DRIVING SIMULATORS

**Ghasan Bhatti[123], Roland Bremond[1], Jean-Pierre Jessel[2], Guillaume Millet[3]
Fabrice Vienne[1]**

[1]IFSTTAR, LEPSIS, France
[2]IRIT, University of Toulouse III, France
[3]OKTAL SA, 19 Boulevard des Nation Unies, 92190, Meudon, France

Implementing the experimental protocol on driving simulators is quite critical to the behavioural researchers, because of the high-fidelity and technical nature of the Driving simulators. Scenario modelling is one of the difficult task in the experimental protocol, where researchers have to manipulate the real-world situations on driving simulators, using a scenario modelling programming language. Researchers are not equipped with certain technical and programming skills, so that they could model scenarios by themselves. In many cases, researchers have to depend on the technical support personnel of the driving simulators, who will model scenarios for them. Also the interactive environment to program scenarios is not so user-friendly. This paper summarizes the needs and requirements of driving simulators users, when they interact with the driving simulators in order to model the scenarios . To gather the user requirements , we conducted interviews of Driving simulator users with different backgrounds. Interviews were conducted under four different themes, that incorporates, General problems, Concept to script translation problems, Ideal approach for scenario modelling and their ideas and feedback for the improvement of riving simulator software in use. From our results, our future work includes the development of a User-Centered Design (UCD) to model scenarios on Driving Simulator software.

Introduction

Driving Simulator is a useful research tool for behavioural researchers to study drivers behaviours, to analyse road safety features, and for drivers training without any safety risk. In order to retrieve the data (to be analysed), researchers have to design and implement experimental protocol on the driving simulators. Executing the experimental design is a quite critical component in any research field (Romano and Stoner, 2007). It requires careful considerations and quite controlled environment(depending on the research objectives) in order to achieve the desired goal of the experiment. To execute the experimental design on driving simulators, researchers have to model scenarios on driving simulators using a programming language environment provided by the Driving simulator software.

In the context of driving simulation, scenarios are termed as , everything that happens in the driving simulator (Papelis et al., 2003). Scenario incorporates specifying and controlling the attributes Ambient traffic and its attributes, ambient environment and simulation conditions, route of the research participants and their position, series of the real world situations which are to be manipulated on driving simulators. And these all parameters and specifications are controlled by programming or scripting language offered by driving simulation software.

Programming the scenarios is a quite challenging task, especially for the researchers, who do not have good programming background. In order to program the scenarios on driving simulators, researchers need to have some programming skills, for which they are not usually trained. And In most of the cases, they have to depend(totally or partially) on the scenario developers or technical persons, who program scenarios for them on driving simulators, which is quite time consuming for them, because they have to depend on other persons. Also the dependency on technical persons can have an issue of communication. So researchers and developers need to develop a common language for discussing the techniques and requirements to manipulate the certain situation. (Papelis et al., 2003). So In order to develop this common language or to avoid this communication issue, It is better for the researchers, that they program scenarios by themselves.

One of the reasons, why programming is difficult, because, programs are abstracts (Cypher and Smith, 1995). especially the people who do not have a solid background, it is difficult for them to think in an abstract manner to implement the same situation, as compared to they think about the situation in the real world. For example, In order to create a certain accident situation, where a car overtakes the subject car(Simulator vehicle) and stop it in front of the subject.

But in order to implement the same situation using a modelling language, there are many parameters to control and take care of, for example, speed and lane position of overtaking car, speed of the subject driving simulator vehicle. Distance to stop from the subject vehicle. So abstraction is one of the barrier while learning programming, especially for the novice users. Also It is common observation, that, the more complex the traffic situation, that users have to manipulate , the more abstraction it is required in the program. Also programming languages have been designed without careful attention to human-computer interaction issues (Newell and Card, 1985), which has made programming activity may be more difficult than necessary(Pane et al.,2001). But there are different design techniques and solutions(Cypher,1991, Pane et al.,2002), which have been proposed in the past two decades, which has really made significant improvement for the non-programmers to get along well with the programming languages

Although there have been significant research in the scenario modelling languages and currently available tools are quite powerful in modelling almost all kinds of situation and manoeuvres. Different techniques and methods have been proposed in order to model scenarios on driving simulators(Allen et al.,2001,6!!!!Papelis et al.,2003, Allen et al.,2004, Krueger et al.,2005, Suresh & Mourant, 2005). But there is no significant contribution on the ease of using these tools and techniques to model scenarios on driving simulators. Also The interaction environment of these tools do not make the scenario modelling task easy and comfortable for the users, and researchers find it difficult to model scenarios by themselves using these tools. Because they were not designed by keeping in mind the user interactions needs of using these scenario modelling languages.

So, there is a need to design and develop a interactive environment or an interactive approach, which enables user to model scenarios quickly and easily. There has not been much contribution to develop a scenario modelling language by keeping in mind the current needs of the users, while modelling scenarios. This paper summarizes the needs of the researchers to model scenarios on the driving simulator, which are gathered during the interviews. We have tried to identify the needs of users in different themes, like what are their general problems,the Abstraction problem; when they have to transform the concept into the script, What is the ideal approach for them to model scenarios, and their ideas and feedback for the improvement of driving simulator software in their use. From the results of these interviews, we are developing a prototype by considering the human-computer interaction issues, in order to meet the current needs of the users. And this prototype will be evaluated by the users in order to improve the design at early stage of the development. This design will be implemented on SCANeR Studio by OKTAL SA, France and ARCHISIM by IFSTTAR, France. In the second part of the paper, we shall describe our method of gathering the user requirements . In the 3rd section, we will present our results, which is followed by a conclusion and future work.

Method

As our objective is to develop an easy and user-friendly Interactive approach for the researchers, who do not have good programming background to model scenarios on driving simulator software. So we have used user-centred design approach, which requires users to be involved at the very early stage of the design process. We have used Interview technique of the user-centred design practice. Interviews technique has many advantage over other techniques. It is a direct and an effective way of gathering user needs and requirements by asking open ended questions, and also to understand the user profiles and context of their use.

We have informally interviewed 19 Driving simulator software users with diverse backgrounds, profiles and experience on different driving simulator software. Users were using different Simulation software and had different backgrounds as specified in the Table 1 and Table 2.One user, who was the Post doctorate researcher was unsure about the name of the software, but shared the experience of using driving simulator software. 6 of the 19 users were interviewed in person, 1 user was interviewed using Skype, and remaining 12 users were interviewed on phone. The interviews were not recorded, but we tried to make sure that we don't miss any detail, by already making some formats on note-taking sheet. Average duration of the interview in person was 40 to 45 minutes, while average duration of telephonic interview was 50 to 60 minutes, while the Skype interview took 83 minutes. The participants were not given any incentive to participate in the interviews.

Table 1: Driving simulators

Number of User	Driving Simulator
7	ARCHISIM
6	SCANeR Studio
2	ST SIM
1	STI SIM Drive
1	VTI Driving Simulator
1	Honda Simulator
1	Unsure about the name of software

All the Ph.D. researchers had used driving simulators during their master thesis, and currently using them for their doctorate studies. So they had approximately one and half year experience of using driving simulators. While the post doctorate student had used driving simulator in Ph.D. study. Behavioural researchers had average experience of 3 year for using driving simulators, while software engineers had 2 year experience of modelling scenarios on driving simulator for behavioural researchers. The research engineer had 2 year experience of using driving simulator.

Table 2: User Profile

Number of User	Driving Simulator
7	Ph.D. researchers
8	Behavioural Researchers
2	Software Engineer
1	Research Engineer
1	Post. Doctorate

Results

These qualitative interviews have given us a very good platform to explore user need by discussing in depth about their experience of modelling scenarios on driving simulators. The details of their participants programming language experience is shown in the Table 3. The expertise level of the users for these programming languages was Basic for all languages except software engineers who had good expertise in C++ and Matlab Simulink, and one researcher, who has good expertise in C++ and Matlab Simulink. In the remaining section we shall discuss the users problems, their needs and requirements identified by most of the users.

Table 3: Programming Language Knowledge

Programming Language	Number of Users
C++/Visual Basic/Programming in Matlab Simulink	3
HTML	2
Java/Action Script/Python	1
No knowledge/experience of programming Language	7

General Problems

General problems refer to the problems faced by the user, while using the software in order to model scenarios. The common problems identified by the users are defined as under.

P1. Finding/Selecting the relevant functions in order to perform a specific task.

P2. Sometimes, It is difficult to test the rules before running the simulation. e.g. If a situation has to occur after 25 minute of simulation time, they can only test the situation after 25 minutes of simulation and not before.

P3. Debugging the Script in order to find the error is quite difficult and time consuming.

P4. Syntax of the scenario modeling language is difficult for the users to follow.

P5. Users have to control and program the scenarios by doing programming at quite low level. Which is quite difficult for them.

P6. Navigating between different scenarios numbers, rule numbers or trial numbers is quite fuzzy in order to follow the script, when the script is quite long and complex.

Concept to Script Translation Problems

Concept to script problem refer to the problems, that users face, when they want to manipulate the real-world situation on driving simulator using programming language of driving simulator. According to users, this is the difficult step for them, as they know, they cant do the same way they think. So the main problems identified by users are as under.

P6. Controlling the ambient traffic using the script is very difficult for the users.

P7. Tuning the events. For example vehicle approaching at intersection and pedestrian is going to cross.

P8. Selection of triggers(speed, distance etc.) in order to tune the events.

P9. Reproducibility of the situations needs extensive scripting, and difficult to manage.

Ideal Approach to modelling scenarios

After the user described their problems, users were asked tell us about their ideal approach to model scenarios, and how their problems can be solved. These ideas are described as under.

I1. Drag and Drop situation for a scenario. 10 users proposed the notion of Dragging and dropping the scenario situations. Users thinks, that, In this way, They just have to specify the specific parameter for a specific situation, and not the complete code.

I2. High-level scripts for scenarios. This idea is the same as I1, but when users were asked that, every user has different requirements, so it will be difficult to manage long list of libraries for dragging and dropping, then user specify that, In order to use High-Level script for complex scenarios, there should be a functionality of editing the high-level script at low-level.

I3. Interaction with the world. Users want to interact with the world map while modelling the scenarios.

I4. There should be an image/preview of the work, users are doing, which will help them in taking decisions, that what they want to do, and how will they do it.

I5. It will be more comfortable for the users, if they click on autonomous vehicle and set their rules and behaviours in the scenario.

I6. It would be easy to specify the tasks/events based on a time-line or distance-line of the experiment.

Improvement of the Driving Simulator Software in use

- Interface of the scenario modelling software. All users have used textual interface for scenario modelling, except the user of SCANeR software, which has Graphical User Interface(GUI). So most of the users who are using textual user interface have preferred to use a GUI for the scenario modelling.

- Searching of the relevant functions should be made easier.

- There should be a very user-friendly documentation for how to use the relevant functions, with some examples.

- There should be only one interface in order to configure different files on the driving simulators.

Conclusion and future work

In this paper, we have summarized the requirements of the users(behavioural researchers) to model scenarios on driving simulator. Although we have interviewed the users of different driving simulators. But the requirements we have presented are those which are almost identified by the users of all driving simulators. Although it is difficult to generalize these requirements for all driving simulators, but If we explore these requirements in a contemplative manner, then these problems are quite obvious. Users have identified some other problems as well, which are quite specific to the software they are using, But these problems do not exist in all Driving Simulator software.

Although users have mentioned their problems, but they have also proposed solution in order to solve those problems. Like in last section. If we adopt the ideas I1and I2, then we can solve problems like P1, P3, P4 P7, P8 respectively. We think that, by following the problems, that are identified by the users , and the ideas proposed by users, and applying the appropriate interaction techniques to these ideas, we can provide users with a user-friendly Interface to model scenarios on driving simulators. Our future work includes the development of the prototype based on these ideas, which will be evaluated by the user, and will be improved, which will lead to the final implementation of ARCHISIM and SCANeR.

Acknowledgement

This work was carried out within the framework of Marie Curie Initial Training Network(ITN) framework for the project “ADAPTATION” funded by the EC

(Grant Agreement number: 238833). we are thankful to the participants, who took part in the interviews.

References

Allen, R.W., Park, G., Rosenthal, T.J., Aponsom B., 2004, *A Process for Developing Scenarios for Driving Simulations.* Paper presented at IMAGE 2004 Conference Scottsdale, Arizona.

Allen, R.W., Rosenthal, T.J., Aponsom B., Parseghian, Z., Cook, M. and Markham, S., 2001, *A Scenario Definition Language for Developing Driving Simulator Courses.* Proceedings of the Driving Simulation Conference, Sophia Antipolis (Nice), France.

Cypher, A. 1991, Eager*:Programming Repetitive Tasks by Example*,Proceedings of ACM CHI '91, New Orleans, New York, USA, 33-39

Cypher, A. and Smith, D. 1995, *KidSim: End-user programming of simulations*,Proceedings of ACM CHI '95, 27-34

Krüger, H.-P., Grein, M. & Kaußner, A. 2005. SILAB – A task-oriented driving simulation. Paper presented at the Driving Simulation Conference, Orlando, Florida

Newell, A., and Card, S. K. 1985. *The Prospects for Psychological Science in Human-Computer Interaction.* Human-Computer Interaction, 1(3), 209-242.

Pane, J.F., Myers, B.A., and Miller, L.B 2002, *Using HCI Techniques to design a More Usable Programming System.* Proceedings of 2002 IEEE Symposia on Human Centric Computing Languages and Environments, Arlington, VA , USA

Pane, J.F., Ratanamahatana, C.A. and Myers, B.A. 2001, *Studying the Language and Structure in Non-Programmers' Solutions to Programming Problems.*

International Journal of Human-Computer Studies 54(2), 237-264

Papelis, Y., Ahmad, O. and Watson, G. 2003, *Developing Scenarios to Determine effects of Driver Performance:Techniques for Authoring and Lessons Learned.* Proceedings of the Driving Simulation Conference,North America, Michigan, USA

Suresh, P. and Mourant, R.R., "A *Tile Manager for Deploying Scenarios in Virtual Driving Environments*". Proceedings of the Driving Simulation Conference North America, December, 2005, pages 21-29