ARTICLE IN PRESS

Transportation Research Part C xxx (2014) xxx–xxx

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Design and evaluation of a user-centered interface to model scenarios on driving simulators

Ghasan Bhatti [a,b], Roland Brémond [a,*], Jean-Pierre Jessel [b], Nguyen-Thong Dang [a], Fabrice Vienne [a], Guillaume Millet [c]

[a] Université Paris Est, IFSTTAR, LEPSIS, Marne la Vallée, France
[b] Université Paul Sabatier, IRIT, Toulouse, France
[c] Oktal SA, Meudon, France

ARTICLE INFO

Article history:
Received 23 January 2014
Received in revised form 1 August 2014
Accepted 17 September 2014
Available online xxxx

Keywords:
Driving simulators
Scenario modeling
User-Centered Design
User interface

ABSTRACT

Modeling scenarios on driving simulators is a complex and difficult task for end-users because they do not have the skills necessary to program the scenarios. In this paper, we present a user-centered architecture in which we have split the scenario modeling interface into 3 sub-interfaces (Template Builder, Experiment Builder, Experiment Interface) based on the user skill. The concept is tested with a panel of end-users, with fair results in terms of performance and subjective judgment.

1. Introduction

Driving simulators are increasingly used to study driving behaviors, road safety features and to design and evaluate Advance Driving Assistance Systems (ADAS). In order to carry out an experiment, the end-user (e.g. behavioral researcher, psychologists, human factor experts) have to prepare an experimental protocol, which involves scenario authoring (e.g. modeling traffic situations and vehicle maneuvers). Such a preparation requires technical and programming skills for which behavior researchers do not have any formal training in most cases. As a result, they depend on technical persons or scenario developers in their respective organizations, which in most cases is a time-consuming and frustrating process, due to communication problems between people with different backgrounds. Moreover, this is also a time-consuming task for the technical team.

A scenario in virtual reality applications typically involves a sequence of actions or events. In the context of driving simulators, a scenario can be regarded as "everything that happens in the simulator", which include specifying and controlling the ambient traffic and its attributes, the ambient environment and the simulation conditions, the route of the participants and their position, the traffic situations and other vehicles maneuvers Papelis et al. (2003). Some authors include both the layout (road network, terrain, driving environment: city, motorway, etc.) and the activities during the experimental trials (critical situations, vehicle maneuvers, etc.) in the concept of "scenario". Some other use the term "scene" to specify the layout and the term "scenario" to specify the activities during the trials (Kearney and Timofey, 2011). In the following, scenario refers to the specification of all activities including critical events, vehicle maneuvers and environment changes.

* Corresponding author.

http://dx.doi.org/10.1016/j.trc.2014.09.011
0968-090X/

In order to develop scenarios, end-users take technical and programming help from the technical persons and the scenario developers. There is a communication gap between the end-users and technical persons, as there is no standardized language or communication method between researchers and technical persons, while their technical background is often very different (e.g. psychologist vs. computer scientist).

One of the reasons why programming is a difficult task for end-user is that programs are abstract (Cypher and Smith, 1995). It is difficult for end-users to think in an abstract manner to implement the specific task rather than thinking about the situation in the real-world.

Every driving simulator provides a scripting language and a scenario authoring tool to develop scenarios. These scripting languages are powerful enough to develop any kind of driving scenario but they do not take into account the skills of the end-users: they are at quite low-level, and they have not been designed by addressing the issues of Human–Computer Interaction (HCI) (Newell and Card, 1985). According to Pane et al. (2002), user interface is one of the factors which can make programming a difficult activity for the end-users. The End-User Development (EUD) systems should fulfill the goals of end-users by taking their skills and objectives into account. In driving simulators, scenario modeling is a challenging task for the end-user because of two additional factors Kearney and Timofey (2011). First, the driving behavior is complicated and not well-understood, so it is challenging to create a reproducible and realistic traffic. The second factor is the variability of the driving behavior.

Thus, there is a need to develop a standardized and user-centered scenario authoring environment that enable and empower the end-user to develop scenarios using the skills they have. In this paper, we present a detailed user-interface (UI) that we have developed using the User-Centered Design (UCD) approach. The remaining sections of the paper are organized as follows. In the next section, we present the related work, which is followed by the details of our method. Then we propose the new interface and its evaluation by a panel of end-users, which is followed by a conclusion.

2. Related work

Significant work has been done to develop usable programming and authoring systems for the end-users in the past, for example Agent sheet (Ioannidou, 2003), Alice (Conway et al., 2000), Programming by Example (Cypher, 1991), etc. Every driving simulation platform provides different user interfaces for scenario authoring which range from textual to graphical format. They also differ in the way critical events and the traffic is specified.

For instance, ARCHISIM (Espie et al., 1994), developed by IFSTTAR, uses textual statement described in a text editor (notepad-like) to specify scenario objects and critical events. They use if-else statements in textual format. The traffic and the critical events are specified in different text files. The SCANeR Software (Reymond et al., 2000) developed by OKTAL SA, uses a graphic way to model scenarios. Scenario objects (Vehicles, Traffic signals, etc.) are placed using the mouse, and scenarios are modeled using Condition-action pairs (If-else statements) in the authoring environment. A more recent example is STISIM (Park et al., 2011) developed by System International, which uses textual statements in their SDL (Scenario definition language). Scenario objects and critical events are described using SDL.

Wassink et al. (2005) proposed a movie set metaphor to generate scenarios dynamically based on Green Dino Virtual Realities' Dutch Driving Simulator. They have proposed the movie set as a driving simulator, where actors (vehicles, pedestrians, etc.) come at the scene and play certain set of roles, which are assigned to them in the script. They have also emphasized on the problem of end-users to model scenarios using a scripting language. A tile-based approach is also used to specify scene and scenario elements in the driving simulator. The world (scene) is divided in tiles, which are configured, assembled and then loaded into the driving simulator during the experimental trial. A "tile" is a section of the route which contains elements like roads, traffic signals, buildings, trees and scenario objects. They are grouped together and loaded using an interface or specifying the tile sequence. In some systems, tiles are static and may not be altered or moved during the experiment run (Papelis et al., 2003), while in other systems tiles as well as data about the tiles (scene objects, scenario objects) can be altered dynamically during the experiment run (Suresh and Mourant, 2005).

3. Method

We have used the User-Centered Design (UCD) approach to model scenarios. UCD methods claim to provide an end-product, which satisfies the users and enable them to achieve their goal while taking their profile into account. For this, we first conducted a user survey, to understand usability problems of existing driving simulator softwares. From this survey, a new approach was proposed and a prototype software has been built. Finally, the proposed prototype was evaluated with a new panel of end-users.

3.1. User survey

Nineteen driving simulator users were interviewed. They had an average of 2 years experience working with driving simulators, and 7 out of 19 had no programming experience at all. In their previous experience, these users always needed help (total or partial) from the scenario developers. For more detail about the survey see Bhatti et al. (2011).

Please cite this article in press as: Bhatti, G., et al. Design and evaluation of a user-centered interface to model scenarios on driving simulators. Transport. Res. Part C (2014), http://dx.doi.org/10.1016/j.trc.2014.09.011

The users explained their problems for the definition and setting of their experimental scenario, and proposed some suggestions for a better use of driving simulators. The main problems shared by most users were:

1. to control ambient traffic around drivers during the critical situations;
2. to be able to tune or optimize the critical events;
3. to find the relevant functions in order to design an event; and
4. the availability of triggers allowing modeling critical events.

The users also suggested some ideas in order to improve the scenario design: to drag and drop critical situations at high-level from a database, to be able to interact with the scene, or to provide a preview of the successive steps needed in order to design an experimental protocol. From this user-survey, seven steps were identified in order to develop an experimental protocol: terrain/scene selection; Configuration of the participant's vehicle; configuration of the autonomous traffic; configuration of the environment (i.e. daylight, weather); selection of the data to be recorded; construction of the critical events; and experiment execution. This sequence was a first guess, and the proposed approach finally includes eight steps (see bellow).

3.2. Proposed approach

Based on this survey, we have designed a new user-centered multi-layer programming solution. We have made clear that the design and execution of a scenario needs three main roles: the end-user (e.g. researcher, trainer, etc.), the technical team, and the experiment operator. A multi-layer architecture was proposed, in order to separate the component corresponding to each role. End-users interact with the higher layer, which is closer to the real world, while technical persons interact with the middle layer and the driving simulator manufacturers interact with the lower layer.

Traditionally researchers and technical persons both interact with the middle layer, and develop the scenarios using the API (Application Programming Interface) or the high-level functions (e.g. changing the environment, the speed of the vehicles) developed by the software manufacturer. The API is the set of programming instructions allowing to access software applications. These APIs can be developed using low-level programming languages (e.g. C++) by driving simulator manufacturers (as for SCANeR), or they can be developed by the technical team within a research institution (as for ARCHISIM). In both cases, using APIs for the scenario development, end-users have to use the syntax and the low-level programming primitives, which is difficult when they do not have these skills. This is why we added an extra layer (higher layer) with which the user can develop the scenarios without directly using the API or the low-level primitives.

The scenario authoring interface is split into three sub-interfaces corresponding to the three groups of users identified during the survey: Experiment Builder, Template Builder and Experiment Interface, as shown in Fig. 1 (see also Bhatti et al. (2013)). The "Experiment Builder" corresponds to the higher layer, while the "Template Builder" corresponds to the middle layer. The higher layer is dedicated to researchers: they can customize the parameters of the scenario event templates at high level with minimum set of skills; the lower layer is for the technical persons who develop the templates at low-level.

The Template Builder sub-interface will be used by technical persons in order to design the Graphical Interface (GUI)-based parameterized templates of the scenario events. They will use existing APIs and build high-level events (e.g. vehicle cut-in, traffic jam). The Experiment Builder interface will be used by a end-user with no or very low programming skills. The parameters of the scenario events will be customized, depending on the scenario, at high level. The Experiment sub-interface will be used by a researcher or any other person who operates the driving simulators. It will run the experiment, and if needed the parameters of the templates and collected data may be modified.

3.3. Prototype development

The proposed interface tries to follow the logical steps followed by the end-users when they develop an experimental protocol. The proposed interface was designed with the issues raised by the Survey in mind, which mostly addressed how critical events are dealt with in the course of scenario development. In the proposed architecture, the users can specify the scenarios at higher level, exploiting the low-level primitives of the scripting languages; this addresses all 4 issues above. We have also tried to reduce the complexity by separating the steps encountered by the users. In the templates, the user can specify the parameters according to the situations in the real world rather than using low-level parameters. Also, using zones made the task much easier for traffic tuning, getting rid of unimportant parameters (according to the users point of view). So we did not address the survey's problems in a one-to-one manner; instead, we proposed a series of changes which contribute to the improvement of the interface, with an input on the problems raised by the users. We do not claim that this is the end of the story, and it will now be possible, from the new prototype interface, to directly address these problems. However, this will now be a matter of interface development, more than a research question.

As the focus of our work is on end-users, we present here the prototype for the Experiment Builder. The prototype was build using "Justinmind prototyper", a tool to develop interactive prototypes. The user was guided using breadcrumb navigation Nielsen (2007). The end-users have to follow 8 steps.

Please cite this article in press as: Bhatti, G., et al. Design and evaluation of a user-centered interface to model scenarios on driving simulators. Transport. Res. Part C (2014), http://dx.doi.org/10.1016/j.trc.2014.09.011

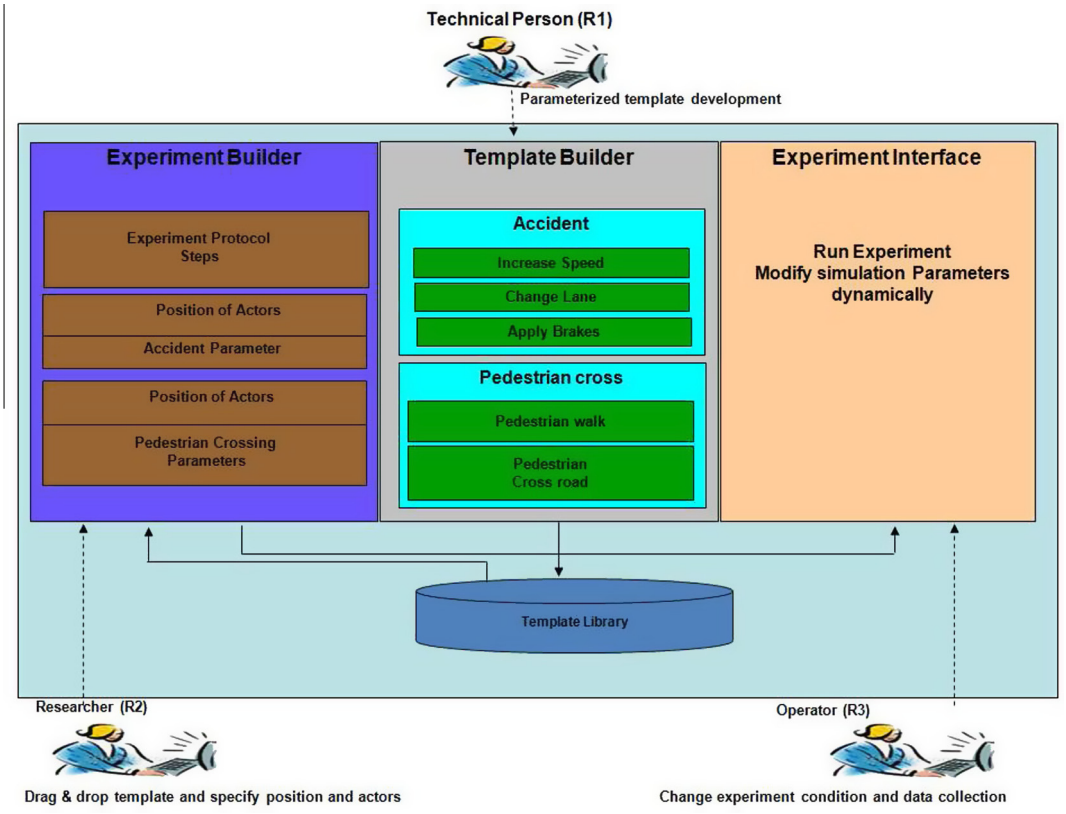

Fig. 1. Experiment building procedure.

Step 1 (Experiment description): the user provides the description of the experiment (name, goal, etc.).
Step 2 (Terrain specification): the user selects a terrain available from the terrain database. He can visualize the selected terrain.
Step 3 (Configuration of the participant's vehicle): the user defines the subject's vehicle. He provides the vehicle model, color, position, and maximum speed. The Itinerary can also be specified, as well as the ADAS.
Step 4 (Traffic zones specification): the user specifies the traffic by creating traffic zones. He specifies the position of each zone on a map, then the traffic density in each zone. He can specify the distribution and behavior of the traffic in each zone. Zones denote any spatial division of the virtual environment, where specific event or context can be defined or tuned. It is a common practice in driving simulator studies to configure traffic with different density in different areas. We propose to let the user define the relevant areas ("zones") in order to specify the traffic according to the planned scenario in an easy way, at high level (instead of a script for each lane, or each portion of each lane, or even individual control of all vehicles in the traffic). The same occur with weather zones: most of the scenario use a global weather tuning, but some may need that fog, or rain, only occur in a specific area, so that the user needs some tools to specify this area on a map.
Step 5 (Environment specification): the user specifies the environment (e.g. lighting, fog, rain, snow). Even through most scenarios use a global weather tuning, some studies need a special weather condition in some local zones. This steps allows defining such local weather zones.
Step 6 (Data collection): the user specifies the dependent variables which will be recorded during the trials, such as speed, acceleration, lateral positioning, and so on.
Step 7 (Specification of critical events): the user creates the traffic situations and vehicle maneuvers (events), which will occur during the trials. They drag templates of the traffic situation or vehicle maneuvers from a template library to the scripting area, and configure the template's parameters (note that these templates are developed by technical persons using the "Template Builder"). After dragging the template in the scripting area, the user will specify what event would trigger it (e.g. a specific position of the subject's vehicle, a specific time).
Step 8 (Spatio-temporal visualization): the user can visualize the whole experiment in temporal and spatial representations. The templates which will be triggered based on time can be viewed on a time-line, while those being triggered based on position can be viewed on a distance-line (Fig. 2).

Please cite this article in press as: Bhatti, G., et al. Design and evaluation of a user-centered interface to model scenarios on driving simulators. Transport. Res. Part C (2014), http://dx.doi.org/10.1016/j.trc.2014.09.011

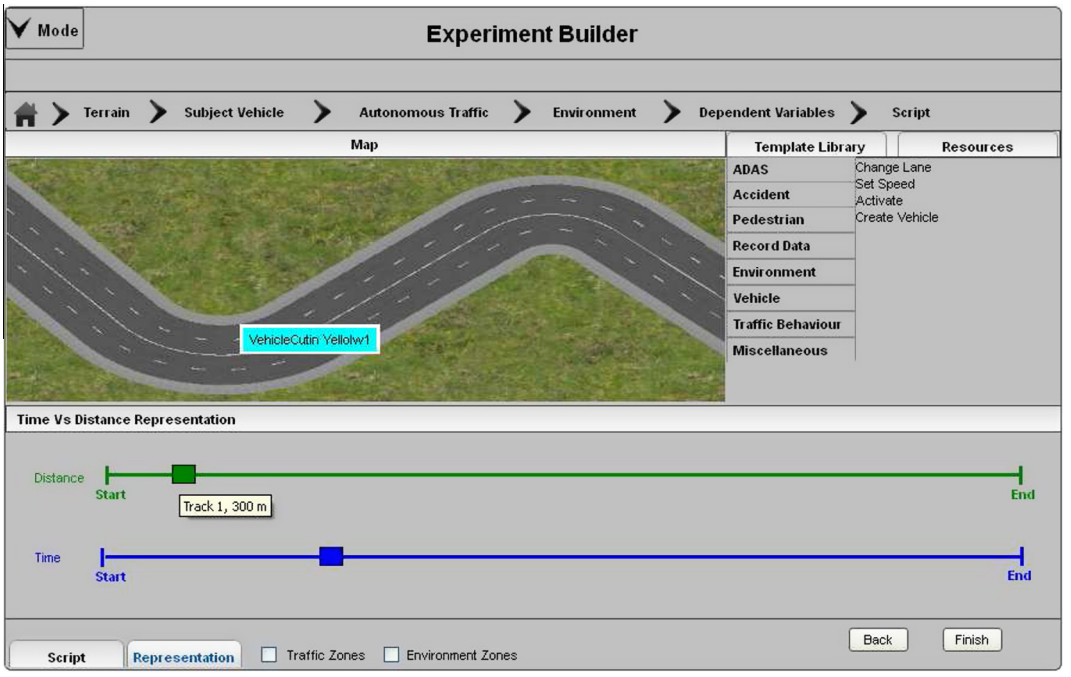

Fig. 2. Visualization of the experiment on a time-line and a distance-line.

4. End-user evaluation

The main purpose of prototypes is to evaluate a system at an early stage of the design. Following a UCD approach, the users were involved in different phases of the design and evaluation of the prototype. This evaluation was split into two phases. A preliminary study was conducted to collect qualitative feedback from a small set of users. Then, a second evaluation was conducted, with more users, to collect both quantitative and qualitative data.

4.1. Preliminary study

A first evaluation was conducted to get the general overview of the proposed approach. Nine users participated in the study. They were behavioral researchers who used driving simulators for their research, but had no programming skills to create scenarios. The proposed approach was first explained, then they performed a small exercise using the prototype. They were observed during the exercise and interviewed later on. In the exercise, they had to create a simple scenario, with a lead vehicle to follow; this vehicle should brake at some time (see Fig. 3). The end-users also had to create a traffic and a weather zone. The average time for a user was 30–45 min with 5–7 min for explanation, 15–20 min for the exercise and 10–20 min for the interviews (the interviews were not recorded).

The structure of UI and the steps to create the experimental protocol were quite clear to all users. They all gave a positive feedback about the approach for creating scenarios/events by using templates. Some minor problems were found: for example, creating a zone for autonomous traffic was not intuitive for four users. Three users also suggested that they should be able to drop the template on the map, in order to create position-based events. During the interviews, the users felt easy and it was intuitive for them to configure the templates, rather than low-level coding, and they were also interacting with the map. Some suggested that knowing the length of the zone can be helpful in creating the experiment. They appreciated the guidance by navigation: a separate sub-interface was provided for each step, minimizing the complexity of creating an experiment.

4.2. Main study

Based on the suggestions by the end-users in the preliminary study, a fully functional tool was developed. A second evaluation was then conducted, in order to get a more quantitative assessment of the proposed approach. Three hypotheses were considered:

1. H1: The proposed method/tool empowers the user to develop an experimental protocol.
2. H2: The end-user will focus on domain problems rather than programming issues
3. H3: The end-user will need lesser or no help using the proposed method/tool.

Please cite this article in press as: Bhatti, G., et al. Design and evaluation of a user-centered interface to model scenarios on driving simulators. Transport. Res. Part C (2014), http://dx.doi.org/10.1016/j.trc.2014.09.011

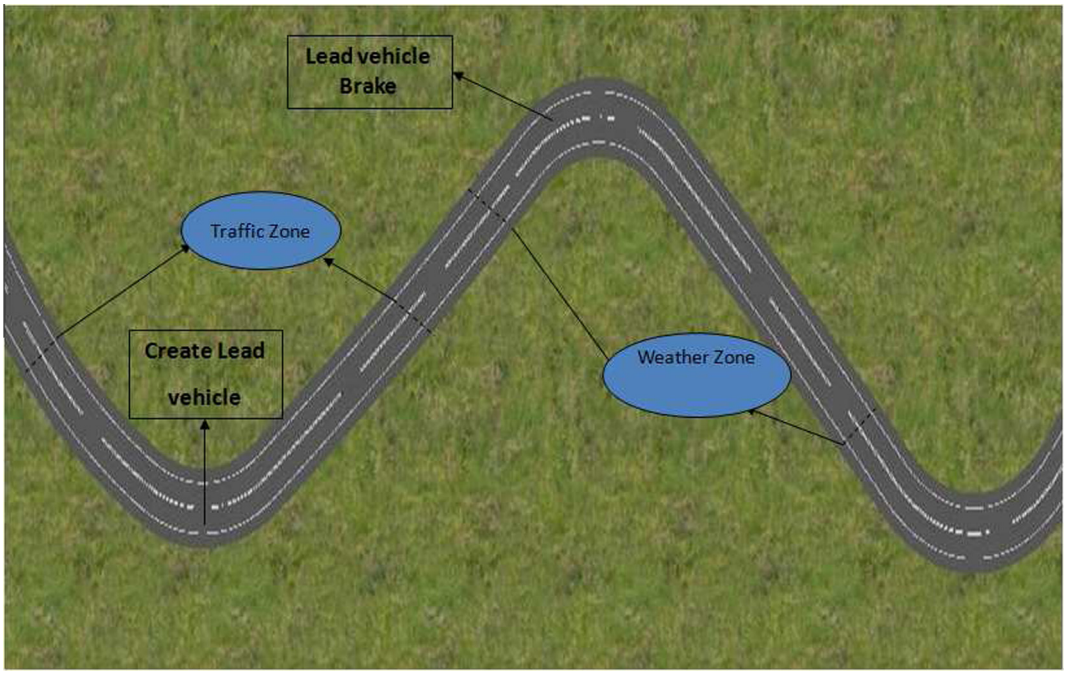

Fig. 3. Exercise to model scenario.

Eighteen participants were recruited for the study. All participants were behavioral researchers who used driving simulators for their research and had an average of four to five years experience of working with driving simulator; they had no programming skills at all. A first group (eight participants) had some experience of developing scenarios on driving simulators, and a second group (eight participants) had no experience. These participants were selected from IFSTTAR and from partners in the ADAPTATION project (www.adaptation-itn.eu). Fourteen used ARCHISIM for their research, two used SCANeR, and the remaining used STI simulator. Nine participants participated in the experiment on site and nine participated online.

For the on-site participants, the computer was equipped with the ActivePresenter software (atomisystems.com/activepresenter), a tool to record screen activity. The participants who participated online used Skype or TeamViewer (www.timeviewer.com).

The participants were first invited to read a tutorial about the scenario development prototype: they could ask any question at this stage. Then, they had to build a driving scenario with the proposed prototype. They were told to think aloud (Van Someren et al., 1994; Lewis and Rieman, 1993) as they were performing different tasks: participants were asked to say whatever they were looking at, thinking, doing and feeling as they went about their task. The scenario was chosen based on a previously published paper (Beggiato and Krems, 2013). This scenario included the features (ADAS selection, traffic zones, weather zones and traffic scenarios) that we wanted to test. The participants then filled three questionnaires: the System usability Scale (SUS) (Brooke, 1996), the Single-Ease Question (SEQ) (Sauro and Dumas, 2009) and a questionnaire related to the tasks performed during the experiment, and were interviewed at the end of their experiment session.

SUS was used as our post-study questionnaire. It has 10 items with 5 choices in each (Likert scale). It can test the usability of almost all kinds of software and hardware applications. It is a reliable and a valid questionnaire which can measure the usability of the system. The shortcoming of this questionnaire is that it cannot identify usability problems encountered by the end-users. In addition to the SUS, we introduced SED as the post-task questionnaire of this study. As its name says, it is a one question questionnaire for each task. It measures the satisfaction of the user while performing a task and can diagnose usability problems.

The simulated road was a 23-km long two-lane highway. With the exception of a 100-km/h speed limit in construction zones, the speed limit was set to 120 km/h. The route consisted of five consecutive road sections of four km each, with two km of straight road at the end. Every road section included two left bends and one right bend, a cut-in situation, one km of construction zone where only the right lane could be used, a lead car (80 km/h) setting the pace, and a queuing situation at the beginning of the construction zone. There were two situations in each zone: a cut-in, and a car-following. The output of the experiment was an XML file, allowing to rate the user performance with respect to a reference XML file. The time each user spent on each sub-task was also recorded.

Please cite this article in press as: Bhatti, G., et al. Design and evaluation of a user-centered interface to model scenarios on driving simulators. Transport. Res. Part C (2014), http://dx.doi.org/10.1016/j.trc.2014.09.011

4.3. Data analysis

We excluded two participants from the analysis since their data was corrupted. As a result, the analysis was performed on two groups of eight participants each, experienced and inexperienced users.

The task performance data was collected from the output XML files. For the experiment description, terrain selection, participant vehicle and data collection steps, if users completed the each subtask it was considered as 100% completed. For the environment zones, if they successfully created the two zones, it was considered as 50% completed (25% for each task). If they successfully configured these zones, they completed the remaining 50%. For the traffic zones, if they created the three zones successfully, it was considered as a 50% completed task (16.6% for each zone). If they successfully configured these traffic zones, they completed the remaining 50% (16.6% for each zone).

The participants had to configure two critical situations (vehicle cut-in and car-following) in each of the three traffic zones. If they successfully configured all cut-in situations, it was considered as 50% completed task (same for the car-following). For the cut-in, they had four templates to configure (they had to turn off an ADAS in the third situation). When they successfully configured the template conditions, it was considered as 25% completed task (6.25% for each template) For the car-following, they had to create a vehicle, and allow it to move: there were six templates in total. If the participants configured the template conditions it was considered as 25% completed task (4.16% for each template).

4.4. Results

All users completed all tasks, except configuring the weather zones and the traffic situations. Only one user could not complete the "Environment configuration" task. To create critical situations, four users completed the task, and 10 users completed more than 75% of the task, one completed 50% and one could not do the task at all.

Configuring the autonomous traffic and creating the traffic situations have been found quite difficult, as expected. Along the situations, it was possible to see whether practice improved performance. Fig. 4 shows a learning behavior, in terms of time spent to the task, for the configuration of the traffic zone and for both cut-in (situation 1) and car-following (situation 2). The learning effect was only observed between the first and second trials, which suggests that using the prototype is quite intuitive.

From the SUS questionnaire, 78% of the users have rated the usability above 70% while 22% rated below 60%. The SUS has been used in many studies, and we could compare these results with other studies. Bangor et al. (2009) associated verbal ratings (between "worst imaginable" to "best imaginable") to SUS score ranges. According to this scale, 6% of the users rated the prototype as "excellent", 72% rated it as "good", 11% as "OK", and 11% as "poor".

The user also filled a questionnaire in which they specified the easiness of each task. All tasks were considered easy by at least 88% of the users; 12% of the users found the autonomous traffic task difficult, and 6% found the environment and critical event tasks difficult. The most interesting point was to compare the performance of experienced and inexperienced users. Task easiness and usability scores were roughly the same among the two groups of users. As expected, experienced users performed better in terms of task achievement: one inexperienced user could not complete the critical situation task at all, and another could not configure the environment zones. Also, the average task completion is better for experienced users.

One interesting feature is about learning. Fig. 5 shows the evolution of task duration across trials, and shows that for each of the 3 tasks (traffic, cut-in and car following), the performance is better for experienced users in the first trial, and nearly the same at the second and third trials, showing that the prototype is well fitted to the unskilled users.

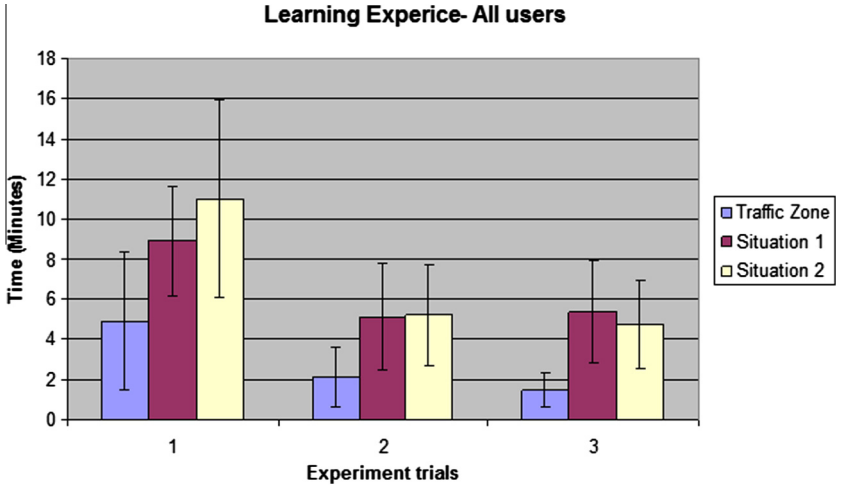

Fig. 4. Learning experience (all users).

Please cite this article in press as: Bhatti, G., et al. Design and evaluation of a user-centered interface to model scenarios on driving simulators. Transport. Res. Part C (2014), http://dx.doi.org/10.1016/j.trc.2014.09.011

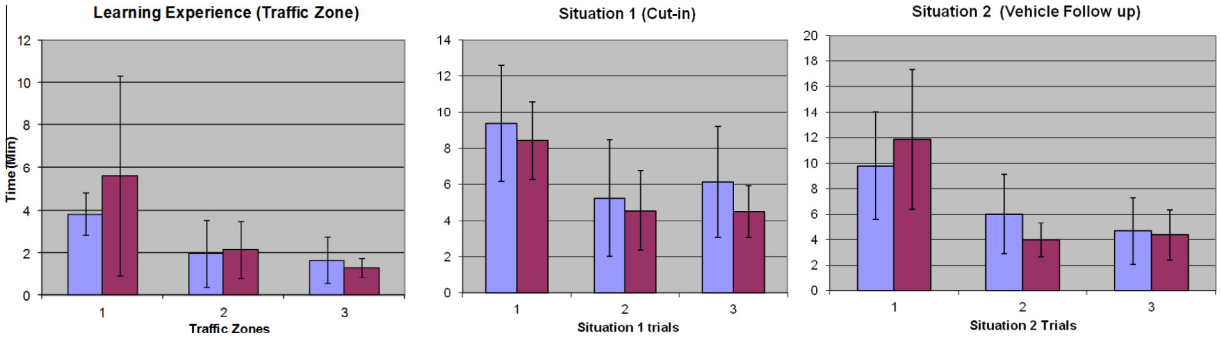

Fig. 5. Learning experience, for each situation. Left: traffic zones; middle: cut-in situation; right: vehicle follow-up. Experienced users are in blue, novices are in red. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Users were asked if any relevant information was missing, about the problems encountered during the exercise, and whether they had any suggestion to improve the interface. One user found there was no information to specify the vehicle as right-hand or left-hand side. Some users raised that the template classification is very important. For example, those related to vehicle movement and overtaking should be in one section. They mentioned that at the start, the software is a bit difficult, but after some practice, it becomes easy (this also appeared in the "learning effect" analysis), so they suggested that at tutorial or training should be provided at the start. The users asked for a documentation of all templates. They also wanted to preview the situation after it is configured. Some users, having worked on other tools, said it was uneasy in the beginning to adapt to this new tool. One encouraging result is that 89% of the users said that would not need help from technical persons anymore using the proposed interface.

4.5. Discussion

The evaluation experiment shows that using the proposed interface, the end-users significantly improved their skills to develop an experimental protocol on driving simulators. Driving simulator end-users usually take total or partial help from technical persons, but using the new approach, most of them could build an experimental protocol without any help. At this stage, they only express the need for a learning step at the start, and a good documentation, which was expected.

The completion rate for inexperienced user was around 60%, vs. 75% for experienced users. The difference is obvious as experienced user have developed experimental protocol before. But the improvement of inexperienced user from 0% to 60% is an interesting figure: those so-called inexperienced end-users had some experience working with driving simulators, but were never able to develop an experimental protocol by themselves.

It would have been interesting to compare the time the end-users took to complete this experimental protocol and the time they would have spent on other systems; unfortunately we were aware of no previous study which could be compared to this one. However, it is clear that most users actually learned how to use the tool. There were still some users who could not complete the critical situation tasks, and it is important to find out why. For instance, some of them did not understand, in the car-following situation, that they had to use 2 templates. This kind of situation may probably deserve some minimal help from the technical staff, but much less than the full development of the scenario.

Back to the three hypotheses above (H1, H2 and H3), our main objective (H1) was to empower unskilled end-users to develop experiments by themselves. People in the "inexperienced" group never developed any experimental protocol, and however managed to complete most tasks. Besides, their average performance was 60% for critical situation, which shows they developed some skills.

Hypothesis H2 was that the end-users would more focus on domain problems. Syntax and abstraction are among the hurdles for unskilled users. In the experiment, participants did not mention any syntactical problem but they emphasized on the lack of template documentation. It is quite logical, since the participants did not have to follow any syntactical rules to create their scenario in this case. All they had to do was to fill the parameters of various templates. To do so, they had to understand the template's parameters and their relevance in a specific situation. The problem is now shifted from programming to understanding different factors in a driving context.

The third objective (H3) was to minimize the role of technical persons, which is expected to benefit to both the end-user and the technical staff. It appears from the results above that this objective is fulfilled, given the high level of autonomy the users achieved, compared to previously.

Our goal was not to raise the end-users autonomy from 0% to 100% in one step. We have demonstrated that changing the programming interface in the domain of driving simulation scenario authoring may give some level of autonomy to the end-users (this also benefits to the technical staff). The gain is from zero (without the proposed interface) to some level between 50% and 100%, depending on the user, for the proposed task. The exact level is not important, as it would depend on each user and scenario complexity. We also could show that learning the new interface was fast, which is among the desirable prop-

Please cite this article in press as: Bhatti, G., et al. Design and evaluation of a user-centered interface to model scenarios on driving simulators. Transport. Res. Part C (2014), http://dx.doi.org/10.1016/j.trc.2014.09.011

erties of an interface. Improvements are still possible, but would probably be a matter of interface development rather than a research question.

In addition, we doubt that an interface could solve all issues that are relevant for scenario authoring, so that we don't think the target level of end-user autonomy is 100%: situations may always emerge where the end-user fails; this is why we have considered some possible feedback towards the technical staff, possibly leading to improvements of the Template Builder.

5. Conclusion

The focus of this paper was on the extension of graphic-based interfaces to a new application domain, namely, scenario design in driving simulators. The contribution is twofold: the domain needs analysis led to the development of a software prototype, making easier the scenario building for people with no programming skills; then, the improvement was assessed with an experiment with participants, using quantitative and objective criterions for the software evaluation. In the end, we demonstrated an improved capacity for end-users with no programming skills, which was the purpose of the new interface concepts.

Following a User-Centered Design approach, we have designed a prototype interface in order to design scenarios for driving simulators. This prototype followed some principles taken from a user needs study (Bhatti et al., 2011): for instance, in the proposed interface, the user's roles, interacting with driving simulators, have been separated. The skills we addressed were in terms of programming skills. We did not want to improve the end-users skills, because most end-users are behavioral researcher, with no background in computer science and do not want to improve these skills. Instead, we propose an interface where these programming skills are not needed any more, so that the end-user may achieve his goal of building the scenario with his own skills. We have divided the experimental protocol procedure into different sub-tasks corresponding to the way users develop their experiment. We have also provided some support at each step using different interaction techniques. The customization of the template should enable the end-user to develop his experiments without following a typical low-level programming process.

Then, this prototype has been evaluated on a panel of end-users. The main objective of this work was to give unskilled users a tool to build their experiment by themselves, as much as possible, instead of asking the technical team. The prototype was evaluated on a panel of unskilled and low-skilled end-users, showing good results in terms of completion rate. Also, the subjective evaluation was encouraging, and we could show that the users could easily learn how to use the tool, so that after 3 trials, the performance of low skilled and unskilled users was roughly the same.

The proposed approach is not platform dependent, and should now be implemented on several driving simulator platforms. We hope it would be beneficial for both the behavioral researchers and the technical teams working on driving simulators.

Acknowledgments

This work is carried out within the framework of Marie Curie Initial Training Network (ITN) for the project "ADAPTATION" funded by the EC (Grant Agreement number: 238833).

References

Bangor, A., Kortum, P., Miller, J., 2009. Determining what individual SUS scores mean: adding an adjective rating scale. J. Usabil. Stud. 4, 114–123.
Beggiato, M., Krems, J.F., 2013. The evolution of mental model, trust and acceptance of adaptive cruise control in relation to initial information. Transp. Res. Part F 18, 47–57.
Bhatti, G., Brémond, R., Jessel, J.P., Dang, N.T., Millet, G., Vienne, F., 2013. Filling the user skill gap using HCI techniques to implement experiment protocol on driving simulator. In: Proc. 6th International conference on Advances in Computer–Human Interactions (ACHI), Nice (France). pp. 401–406.
Bhatti, G., Brémond, R., Jessel, J.P., Millet, G., Vienne, F., 2011. User requirements to model scenarios on driving simulators. In: Proc. 5th International Conference on Driver Behaviour and Training (ICDBT), Paris (France). pp. 493–499.
Brooke, J., 1996. SUS. A quick and dirty usability scale. In: Usability Evaluation in Industry. pp. 189–194.
Conway, M., Audia, S., Burnette, T., Cosgrove, D., Christiansen, K., Deline, R., Durbin, J., Gossweiler, R., Koga, S., Long, C., Mallory, B., Miale, S., Monkaitis, K., Patten, J., Pierce, J., Shochet, J., Staack, D., Stearns, B., Stoakley, R., Sturgill, C., Viega, J., White, J., Williams, G., Pausch, R., 2000. Alice: Lessons learned from building a 3d system for novices. In: Proc. CHI 2000. pp. 486–493.
Cypher, A., 1991. Eager: Programming repetitive tasks by example. In: Proc. SIGCHI conference on Human Factors in Computing Systems: Reaching Through Technology. pp. 33–39.
Cypher, A., Smith, D.C., 1995. Kidsim: end user programming of simulations. In: Proc. SIGCHI Conference on Human Factors in Computing Systems, Colorado, USA. pp. 27–34.
Espie, S., Saad, F., Schnetzler, B., Bourlier, F., Djemane, N., 1994. Microscopic traffic simulation and driver behaviour modelling: the ARCHISIM project. In: Road Safety in Europe and Strategic Highway Research Program. pp. 22–31.
Ioannidou, A., 2003. Programmorphosis: a knowledge-based approach to end-user programming. In: Bringing the Bits Together, Ninth IFIP TC13 International Conference on Human–Computer Interaction, Zurich, Switzerland.
Kearney, J.K., Timofey, G., 2011. Scenario authoring. In: Fisher, D.L., Rizzo, M., Caird, J., Lee, J.D. (Eds.), Handbook of Driving Simulation for Engineering, Medicine, and Psychology. CRC Press/Taylor and Francis, Boca Raton, FL.
Lewis, C., Rieman, J., 1993. Task-centered user interface design. A Practical Introduction.
Newell, A., Card, S.K., 1985. The prospects for psychological science in human–computer interaction. Hum Comput Interact, 209–242.
Nielsen, J., 2007. Breadcrumb navigation increasingly useful. Jakob Nielsen's Alertbox.
Pane, J.F., Myers, B.A., Miller, L.B., 2002. Using HCI techniques to design a more usable programming system. In: Proc IEEE Symposium on Human Centric Computing Languages and Environments. pp. 198–206.

Please cite this article in press as: Bhatti, G., et al. Design and evaluation of a user-centered interface to model scenarios on driving simulators. Transport. Res. Part C (2014), http://dx.doi.org/10.1016/j.trc.2014.09.011

Papelis, Y., Ahmad, O., Watson, G., 2003. Developing scenarios to determine effects of driver performance: techniques for authoring and lessons learned. In: Proc. Driving Simulation Conference North America, Dearborn, Michigan.
Park, G.D., Allen, R.W., T., Rosenthal, J., 2011. Flexible and real-time scenario building for experimental driving simulation studies. In: Proc. CHI 2011, Palo Alto, CA.
Reymond, G., Heidet, A., Canry, M., Kemeny, A., 2000. Validation of renault's dynamic simulator for adaptive cruise control experiments. In: Proc. Driving Simulation Conference (DSC00), Paris, France. pp. 181–191.
Sauro, J., Dumas, J.S., 2009. Comparison of three one-question, post-task usability questionnaires. In: Proc. SIGCHI Conference on Human Factors in Computing Systems, pp. 1599–1608.
Suresh, P., Mourant, R.R., 2005. A tile manager for deploying scenarios in virtual driving environments. In: Driving Simulation Conference North America, Orlando, Florida.
Van Someren, M.W., Barnard, Y.F., Sandberg, J., 1994. The Think Aloud Method: A Practical Guide to Modelling Cognitive Processes. Academic Press, London.
Wassink, I., van Dijk, E., Zwiers, J., Nijholt, A., Kuipers, J., Brugman, A.O., 2005. Bringing hollywood to the driving school: dynamic scenario generation in simulations and games. Proc. First International Conference on Intelligent Technologies for Interactive Entertainment. Springer-Verlag, Madonna di Campiglio, Italy.

Please cite this article in press as: Bhatti, G., et al. Design and evaluation of a user-centered interface to model scenarios on driving simulators. Transport. Res. Part C (2014), http://dx.doi.org/10.1016/j.trc.2014.09.011