DICE Part Editor: Evolution of an End-User Visual Programming Language

Daniele Dellagiacoma

Department of Information Engineering and Computer Science
University of Trento
via Sommarive 14, 38123, Trento, Italy
daniele.dellagiacoma@unitn.it

Abstract

Nowadays the end-user development (EUD) tools are widely used and in constant evolution. They aim at helping especially non-technical users to create or modify own software artifacts without a deep knowledge of programming language. In this paper I examine the development of the DICE Part Editor (DPE), which is a EUD tool used to customize the artificial intelligence (AI) of the non-player characters (NPCs) through the creation of complex behavioral models. I analyze the changes that the DPE went through from its first evaluation with the end users to the current situation.

1. INTRODUCTION

End-user development (EUD) refers to tools, activities and methods that allow end-users, hereby defined as non-professional software developers, to create or modify a software artifact without a deep knowledge of programming language behind it [1].

It is known that programming languages are difficult to learn and use [2], and sometimes require skills that many people don't have. For this reason one of the goals of EUD is to balance the works between professional and non-professional programmers.

In the last years the number of end-user programmers has increased more and more, in fact nowadays there are more end-user programmers than professional programmers [3]. Moreover it has even been perceived that non-programmers can create quite complex programs with a little training [4], depending on the tool. In addition nowadays EUD aims at assisting professional programmers too, for example during the testing phase or to support learning.

An example is the use of graphics as medium to show information about the program state, which can be more effective than purely textual display. In fact, the human visual system is certainly optimized for multi-dimensional data rather than one-dimensional text, which does not exploit the full power of the brain [5].

A visual representation can provide further advantages during the development phase, such as an easier development thanks to less concepts required and a visual feedback that show in real time what the users are doing. Other advantages can be to decrease the time of the development process and to improve the precision with which people perform a programming task.

There are two common types of activities that are allowed by EUD [6]. The first is "parameterization and customization", which allows users to choose from some interaction mechanisms already available in the application. The other activity is "program

creation and modification" aims to modify or create a new software.

The DICE Part Editor (DPE), which is part of PRESTO project, belongs to the first category of activities.

2. PRESTO

PRESTO (Plausible Representation of Emergency Situations for Training Operations) started in 2013 as a project financed by Provincia Autonoma di Trento. The project is currently under development by Delta Lab [7], the Research and Development division of Delta Informatica Spa, a software house located in Trento.

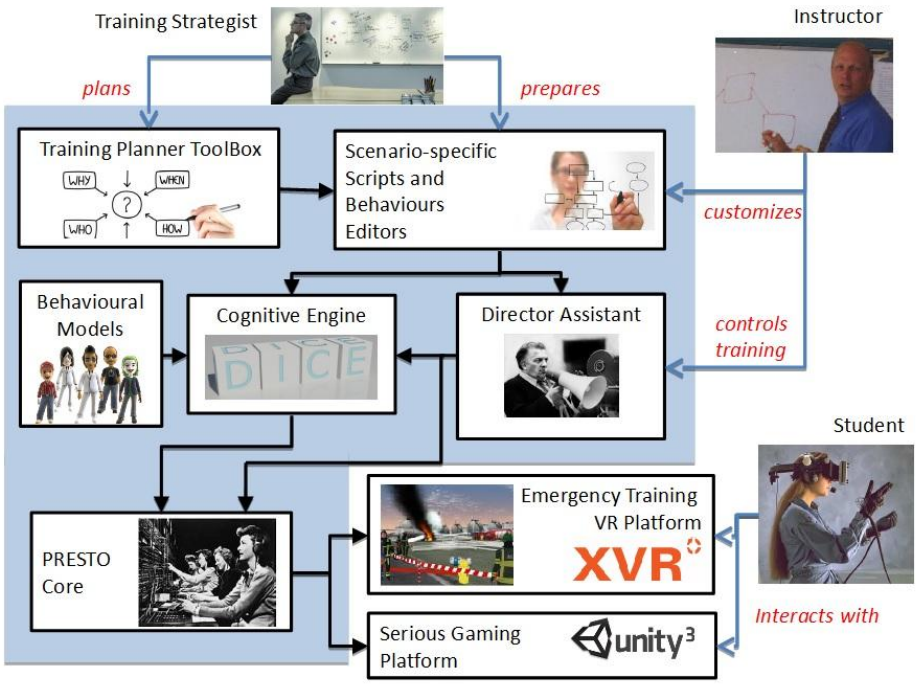

Figure 1: how PRESTO project works.

PRESTO aims at developing a serious game for emergency training and improve this virtual simulation with non-player characters (NPCs) that act in autonomy according to the needs of the trainer. While the trainee plays inside the virtual simulation, using his/her skills and knowledge to complete a set of specific goals, the trainer supervise and direct the training session (Fig. 1).

Furthermore, PRESTO's goal is to provide EUD tools to the trainer to allow the personalization and the management of the simulation and the artificial intelligence (AI) of NPCs.

3. DICE Part Editor

One of the EUD tools of PRESTO is the DICE Part Editor (DPE), which aims at creating or modifying the behavioral models of the NPCs populating the virtual reality [8]. The DPE has been designed for the training strategist, a person between programmer and trainer. His/her work is to prepare and plan scenarios and behavioral models that will be used during the training. The behavioral model of a NPC defines how the character will act in specific situations.

The main structure of the DPE is represented by a grid, where the user can drag-and-drop a goal from the left toolbox to a specific cell. Goals represent the smallest and simplest action that NPCs can carry out during the training session. More than one goal in a single cell means that set of goal will be achieved simultaneously.

The DPE is provided with four basic control structures – sequence, condition, loop and interruption – which can be combined to create new elaborated behavioral models (Fig. 2). The interaction constrains the actions of the users: the usability increases dramatically with a reduced margin of error though the

expressiveness of the visual language decrease.

Zeno Menestrina [9], who has been a Master degree student in Computer Science, has started the development of the DPE in collaboration with Delta Informatica. In a second time I joined the PRESTO project starting to work on the DPE for my thesis [10] in Academic year 2014/2015.

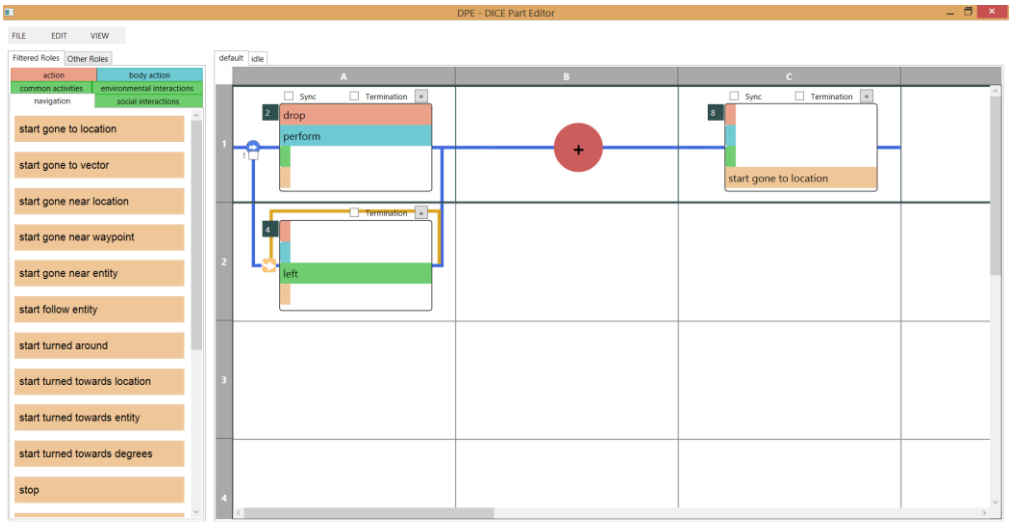

Figure 2: the DICE Part Editor interface.

The DPE development went through several changes during the course of time, which have been necessary to follow the evolution of PRESTO. Also the target of the DPE has been changed, it switched to the trainer from the training strategist, who has more skills in computer science and is able to create behavioral models more complex.

4. EVALUATION

After this first development phase, the DPE went through a testing and evaluation phase. This phase took place during the course of Agent-Oriented Software Engineering (AOSE), part of the master in computer science at the University of Trento and held by professor Paolo Giorgini, which occurred during the second semester of the Academic year 2014-2015. The participants were chosen based on their programming background, similar to the one of the training strategists.

In a first phase, some DeltaLab employers provided an overview of the PRESTO project and fundamentals of AIs. The overall goal of the project was the development of some NPCs' behavioral models through the DPE and test them inside a virtual simulation named DADI.

The phase of collecting feedback and advices was made during four laboratory lessons and two focus groups. The students that took part in laboratory lessons were 15, instead students that participated to focus groups were 11.

This allowed a dynamic re-design of the interface and seven updates of the DPE. To simplify the error recovery one of the first functionalities implemented was “Undo/Redo”, it has been suggested from the users and allows to come back to previous states of the systems and to cancel a specific number of changes. Its implementation has been necessary to provide more control of the system for the user.

It has been added a color transition to the first row/column in case of MouseOver to underline the interactivity of the grid structure, in fact the user can delete an entire column instead one cell by one. A row can’t be delete because it would destroy the all created structure.

Adding a pop-up is another relevant update that has been done. When the user’s saving

the project, the pop-up displays the parameters that the user forgot to specify. This feature is useful to prevent some errors and to inform users about values that he/she didn't specify.

There were other comments and advices from users that weren't implemented in the current version of the DPE yet but they can represent possible future developments.

Users required more shortcuts (i.e. CTRL + C, CTRL + V) and a research function for the ontologies to simplify navigation to expert users. Other features required by users are the ability to drag&drop elements that are already present in the grid and a tooltip that summarizes the selected values of the condition.

However, during two weeks it has been proved the simplicity of interaction with the DPE by students of AOSE though it needs a deep knowledge of AIs. The addition of more and more functionalities allowed more rich language even though the usability of the DPE decreased and end-users need deeper programming and AIs knowledge.

5. DESIGN

Almost every design choice of the DPE was based on the literature and on the Jakob Nielsen's heuristics [11], furthermore they could be used to fix some problems that are still present.

The addition of a pop-up to notify user of what parameters he/she forgot to specify, is based on the ninth Nielsen's heuristic which declares that the user should be helped to recognize, diagnose and recover errors. The function "Undo/Redo" is instead based on the fifth heuristic which allows the user to go back to previous states of the system and prevent errors.

One problem that is still present concerns the semantic of the roles and goals in the DPE. The problem has been underlined by users who don't understand the difference between some goals (i.e. the difference between "slept" and "start sleep"). This problem can be related to sin "More is more" of the seven deadly sins [12].

Another sin of the design of the DPE underlined by users is its visual structure. Users reported problems in seeing the end of a loop when many steps are included because the loop is displayed only when the mouse cursor is over it. Even when there are more steps nested, users can't see the entire behavioral model on which they're working.

6. CONCLUSIONS AND FUTURE WORKS

Even though the DPE went through several updates and changes during the time, there are still some problems that influence negatively

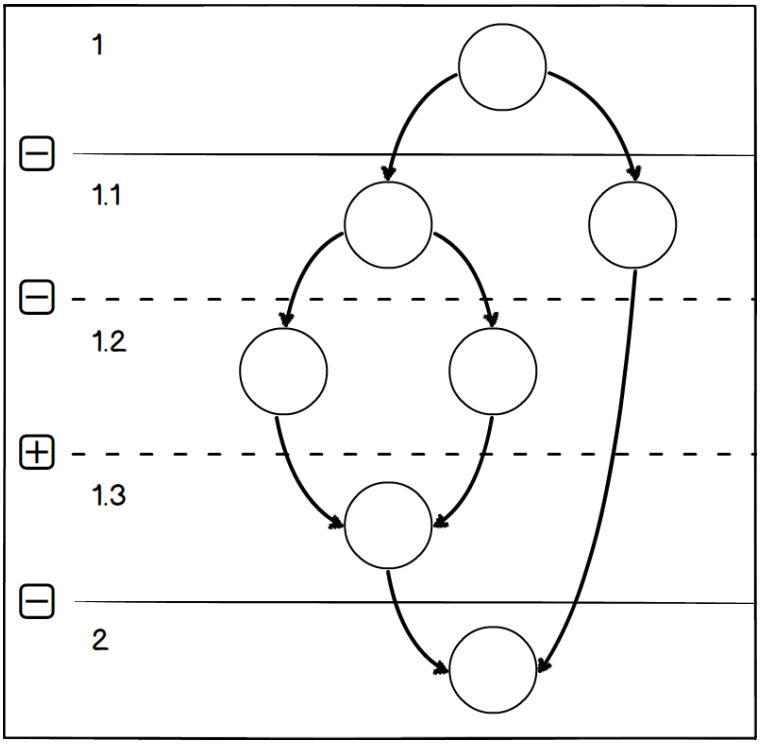

Figure 3: a mockup of a possible different visual representation of the DPE.

the entire interaction with the DPE.

One of the main problem of the DPE is its visual structure, the solution of this problem is strongly related to the end users because it's not completely clear who will they be and the skills that they should have.

If the end user of the DPE would be a professional developer with advanced competence in programming and AIs, the visual structure of the DPE could be shifted from a flow chart to a graph with multiple layers (Fig. 3). This structure can provide better global view of the project and an easier navigation for expert users.

Other features that can be implemented on the DPE, if the end user is a professional developer, are the ability to insert comments and to see and adjust the XML file while using the DPE.

On the other hand, if the end user of the DPE would be a non-technical user (i.e. firefighters, polices or hospital staff), the visual structure would change completely. It should be easier to learn and use though the language expressiveness decrease.

In this paper it has been proposed the evolution of the DPE through many changes. The DPE is progressing in parallel with PRESTO project and as long as its target won't be decided, its development can't be completed.

REFERENCES

[1] "End-user development," *Wikipedia, the free encyclopedia*. 2015.

[2] L. Gould and F. William, "Programming by Rehearsal." Xerox Palo Alto Research Center Research Report, Palo Alto, 1984.

[3] C. Scaffidi, M. Shaw, and B. Myers, "Estimating the Numbers of End Users and End User Programmers," in *Proceedings of the 2005 IEEE Symposium on Visual Languages and Human-Centric Computing*, Washington, DC, USA, 2005, pp. 207–214.

[4] D. C. Halbert, "Programming by Example," PhD Thesis, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, 1984.

[5] B. A. Myers, "Visual programming, programming by example, and program visualization: A taxonomy," *ACM SIGCHI Bull.*, vol. 17, no. 4, pp. 59–66, 1986.

[6] H. Lieberman, F. Paternò, M. Klann, and V. Wulf, "End-User Development: An Emerging Paradigm," in *End User Development*, H. Lieberman, F. Paternò, and V. Wulf, Eds. Springer Netherlands, 2006, pp. 1–8.

[7] "Deltalab - Divisione ricerca e sviluppo Delta Informatica." http://lab.deltainformatica.eu/.

[8] Z. Menestrina, A. De Angeli, and P. Busetta, "APE: End User Development for Emergency Management Training," in *2014 6th International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES)*, 2014, pp. 1–4.

[9] Z. Menestrina, "BMCT: a UI for high level orchestration of serious games," MSc Thesis, Department of Information Engineering and Computer Science, University of Trento, Italy, 2013.

[10] D. Dellagiacoma, "Re-design e valutazione di un'interfaccia di end-user development per serious games," Thesis, Department of Information Engineering and Computer Science, University of Trento, Italy, 2015.

[11] J. Nielsen and R. Molich, "Heuristic Evaluation of User Interfaces," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 1990, pp. 249–256.

[12] L. McIver and D. Conway, "Seven Deadly Sins of Introductory Programming Language Design," in *Proceedings of the 1996 International Conference on Software Engineering: Education and Practice (SE:EP '96)*, Washington, DC, USA, 1996, pp. 309–316.