The design of a process for simplifying the insertion of a new scenario into the virtual simulator

Petr Svoboda^{1*}, Jakub Rak¹, and Václav Lošek¹

¹Tomas Bata University in Zlín, nám. T. G. Masaryka 5555, 760 01 Zlín, Czech Republic

Abstract. The research presented in this article is focused on managing the process of inserting new scenario into the virtual simulator. The first part of the article defines the basic logical division of scenarios whilst the next part contains the design of a tool for MS Excel. The tool is used to simplify the addition of a scenario into the existing simulator by means of an automated SQL script generation and this article describes its functionality together with an algorithm of its usage. The conclusion deals with checking the functionality of the tool by running the resulting SQL script with the existing database and by checking that the loading of the database with new scenario is correct.

1 Introduction

Virtual simulators are useful tools for training members of the armed forces all around the world. The simulator enables participants to acquire knowledge and skills in a virtual environment; it also allows a large variety of scenarios while the risk associated with training is minimized. The training of employees of the private security industry is currently being conducted in a standard way that can be referred to as live simulation. There exist virtual military simulators such as Virtual Battlespace 3 but they need to be redesigned and customized to enable training by means of up to date virtual simulators. It means that requirements for the redesign and customisation will have to be specified to software developers. [1, 2, 3]

A document entitled the Software Requirements Specifications, also known as the Requirements Document, is the output from the analysis of customers' requirements. This document can be characterized as an official summary of requirements that the software developers should implement. This analysis is usually done by an IT specialist – an analyst working as a SW developer.

The research presented in this article is focused on facilitating the insertion of a scenario into the existing virtual simulator. It is a follow-up to an earlier research, which was aimed at simplifying the insertion of new object types. [4, 5]

2 Requirements for the implementation of a scenario

The need for implementation of a scenario into the simulator is based on the need for the creation of an arbitrary scenario, i.e. the model situation. Therefore, the objective of the research is the design of a course of

training that is the scenario of the actual simulation. As this task is rather complex, it is divided into two parts – the general and the specific – for the sake of clarity. The general part includes the following entries:

- a) **Title** – defines the name of the implementation.
- b) **Purpose** – defines the purpose of the implementation.
- c) **Relation** – defines the relation of the implementation towards the existing system; the use of charts for illustrating these relations is recommended.
- d) **User characteristics** – contain the characteristics of users, i.e. the group of participants for which the implementation is suitable.
- e) **Preconditions and dependencies** – a list of all factors affecting the requirements included in the Requirements Document. For instance, there is a software requirement for a specific operating system; it means that the operating system is both a precondition and a constraint.
- f) **Requirements specification** – contains comprehensive specifications of the requirements in such detail that they can be implemented by SW developers. See the investigative questions mentioned below.
- g) **Notes** – supporting and related additional information for understanding the Requirements Document.

For a structured design of a specific part of user requirements, basic investigative questions were used. These questions were selected based on the fact they are well known in the field of criminalistics and they belong to the basic knowledge in this field. The list of questions and their description is as follows:

* Corresponding author: psvoboda@utb.cz

Who?

This question helps to define types of live objects that are to appear in the scenario. As standard, these are divided into the following categories:

- a) **An employee of the private security industry (PSI)** – the object usually controlled by a person – a trainee. In scenarios, which include more PSI employees, these objects can be controlled solely by the trainees or by a combination of trainees and the AI.
- b) **Perpetrator** – the object type usually controlled by AI simulator or operator.
- c) **Neutral person** – an entity controlled by the AI in the vast majority of cases.
- d) **Animal** – an entity controlled by the AI in the vast majority of cases.
- e) **Other entity** – an entity or entities that appear after the implementation of a new object type (e.g. policeman, postman, etc.) based on the particular simulator. They can be controlled by the operator or AI.

What?

The answer to this question serves to define tasks of individual entities in the scenario. In other words, it describes activities performed by individual occurrences of object types in the scenario. A description of some typical activities that can be assigned to individual entities is stated below.

- a) **PSI employee** – a general task to be performed by the employee supplemented by some important details if need be.
- b) **Perpetrator** – activities of the perpetrator, which usually lead to the acquisition or destruction of the selected object. Similar to the PSI employee, there is a certain main goal that is to be achieved using individual activities.
- c) **Neutral person** – the task of this entity in the simulation is usually only moving from A to B.
- d) **Animal** – usually performs simple commands given by a controller (entity), while its primary task is the protection of the entity or building and its perimeter.
- e) **Other entity** – if there is another entity, it is necessary to specify its tasks (activities) within the simulation.

When?

The time definition can be crucial for a number of related parameters, in particular those characterizing the surroundings. For instance, time (date) of the ongoing scenario can affect brightness and frequency of occurrence of entities (e.g. the neutral entities), or the occurrence of snow if the parameters are implemented into the simulator.

Where?

Determining the place where the scenario takes place is one of the basic tasks and it is done in the following way:

- a) Individual live and inanimate objects are distributed in space.
- b) The topology of optional buildings and their surroundings is specified.
- c) Other.

It is recommended to use drawings for specifying the topology; for specification of user requirements basic graphic software can be used, for example user-friendly MS Word.

How?

The answer to this question is a list of non-standard actions of individual live objects occurring in the given scenario. Standard actions include walking, running, sidling, crouching, lying down, jumping, pushing, pulling, getting in or out of a vehicle, opening and closing a door and climbing. The objective of this section is to summarize the behaviour of individual computer-controlled objects that are to be found in the scenario.

By what?

This question identifies the material characteristics of individual live objects, or in other words, the equipment available, i.e. the inanimate objects used by live objects to achieve their goals and perform actions. Generally, these can be divided as follows:

- a) **Gear** – weapons intended for close or remote combat. These consist of types of cold-steel/firearms/other weapons and anything generally used for combat.
- b) **Equipment** – this term defines protective and other elements worn on the body. In particular, they serve for protection of health or they have a specific purpose, such as concealing the identity of the perpetrator. In addition, they include both standard and special clothing (gloves to eliminate fingerprints, balaclava helmets to prevent identification of someone's face), protective clothing (bullet-proof vests, gasmasks, etc.).
- c) **Tools** – the tools that primarily serve to overcome perimeter, spatial, and external protection and the protection of objects. This category also includes tools for overcoming mechanical barrier systems (MBS), tools for overcoming alarm security systems and distress systems (ASDS), auxiliary tools (torches, binoculars, ladder, etc.), communicators (walkie-talkies, telephone) and other.

Why?

This question partly differs from the previous ones because it aims to determine the nature of training and outline the expected course of the simulation. The purpose of training is therefore queried and the answer describes the expected method of achieving the set goals. Apart from the questions described above, the Requirements Document also contains the section Notes, in which other details can be specified and further

context explained. A preview of the proposed Specifications form is depicted in Fig. 1.

3 Algorithm for working with the tool

The PSI analyst sets the number of occurrences of individual object types in the scenario in the appropriate sheets of the workbook, and the tool automatically generates the occurrence of the object types in sheet Výchozí (Default); these are then assigned attributes depending on setting the tool X1_ttypy_objektu (X1_object_types). If needed, the PSI analyst can manually change the assignment of attributes in the following step. Consequently, the occurrences of all object types are generated in the sheet Vše (All). For unassigned attributes, the tool further sets the value 0 and then it sets a random value of 1 to 10 for assigned attributes. Subsequently, the CREATE and INSERT scripts are automatically generated. If necessary, the PSI analyst can manually adjust the previous values of parameters to update the CREATE and INSERT scripts. Once these steps are completed, the tool will automatically merge the existing fixed (default) DROP script in the sheet Vše (All) with the scripts CREATE and INSERT; the DROP script removes the existing tables while the script CREATE recreates them and the script INSERT loads the tables with data. In the last step, the PSI analyst copies the generated scripts and, together with the tool X2_scenar (X2_scenario), encloses them to the resulting Requirements Document, see Fig. 3. [3]

4 Work with the tool X2_scenar (X2_scenario)

The following chapter describes the actual work with the previously mentioned workbook X2_scenar (X2_scenario). [3] The created tool consists of 11 sheets of the MS Excel workbook, which represent individual categories, 1 sheet called Výchozí (Default) and 1 sheet called Vše (All). 11 sheets are as follows:

- 1_Živé (1_Live).
- 2_Neživé objekty exteriér (Inanimate objects exterior).
- 3_Neživé objekty interiér (Inanimate objects interior).
- 4_Neživé předměty interiér (Inanimate items interior).
- 5_Neživé předměty exteriér (Inanimate items exterior).
- 6_Neživé předměty oblečení (Inanimate items clothes).
- 7_Neživé předměty nástroje (Inanimate items tools).
- 8_Neživé předměty výzbroj (Inanimate items gear).
- 9_Neživé předměty výstroj (Inanimate items equipment).
- 10_Neživé předměty lup (Inanimate items loot).
- 11_Prostředí (Environment).

The contents of these sheets are based on the contents of the same name sheets of the general tool X0_ttypy_objektu (X0_object_types) described above. Each of these 11 sheets allows entering a value for the definition of the number of occurrences of individual types of objects in the scenario, see Fig. 1. [7, 8]

	A	B	C	D
1	Položky	Počet		
2	Pachatel	2		
3	Neutrální osoba	2		
4	Pracovník PKB	2		
5	Pes	1		
6				

Fig. 1. Definition of the number of occurrences of the category Živé (Live) objects.

Upon entering values into the column *Počet* (Number), the relevant data is merged into the sheet Vše (All) in the following form:

$$\text{typ objektu} + \# (\text{object type} + \#)$$

where:

- *object type* – is the type of objects defined in the appropriate sheet in the column Položky (Entries),
- *#* – is the order number of the object in the event there are multiple identical types of objects in the scenario.

Thus, the sheet Výchozí (Default) lists all occurrences of object types together with assigned attributes, based on the tool X1_ttypy_objektu (X1_object_types). [3] For the purposes of the current scenario, assigning attributes to the different object types can be adjusted in this sheet (see Fig. 2) by simply changing the value:

- o – indicates assignment of an attribute.
- x – indicates the attribute is not assigned.

	A	B	C	D	E	F	G	H	I
1	Položky	Umístění	Velikost	Váha	Přesun	Kondice	Hlučnost	Obrátlost	Poznámka
2	Pachatel 1	o	o	o	o	o	o	o	Specialista na boj zblízka
3	Pachatel 2	o	o	o	o	o	o	o	Specialista na překonání zámek
4	Neutrální osoba 1	o	o	o	o	o	o	o	Kolemjedoucí
5	Neutrální osoba 2	o	o	o	o	o	o	o	Ridič kolem jedoucího vozidla
6	Pracovník PKB 1	o	o	o	o	o	o	o	Průvod, pravidelná obchůzka
7	Pracovník PKB 2	o	o	o	o	o	o	o	Stacionární ostraha na vratnici
8	Pes 1	o	o	o	o	o	o	x	Středně vyvířený pes posluha
9	Automobil 1	o	o	o	o	o	o	x	Kolem jedoucí vozidlo (Neutra
10	Dodávkový automobil 1	o	o	o	o	o	o	x	Vozidlo určené pro lup, řízení
11	Sloup 1	o	o	o	o	x	x	x	Jeden ze sloupů patřících k obj
12	Sloup 2	o	o	o	o	x	x	x	Jeden ze sloupů patřících k obj
13	Sloup 3	o	o	o	o	x	x	x	Jeden ze sloupů patřících k obj
14	PIR 1	o	o	o	o	o	x	x	Běžný pohybový detektor s výř
15	PIR 2	o	o	o	o	o	x	x	Běžný pohybový detektor s výř
16	PIR 3	o	o	o	o	o	x	x	Běžný pohybový detektor s výř

Fig. 2. X2_vyskyt_objektu (X2_object_occurrence) – the sheet Výchozí (Default).

The last sheet is the sheet Vše (All), see Fig. 4. As standard, the first row contains the headers of individual columns, while the second row of this sheet is to be loaded with data types set in the tool

X1_typy_objektu_vlastnosti
(X1_object_types_characteristics).

Fig. 3. Algorithm of the tool X2_scenar (X2_scenario) used to define occurrences of object types in the scenario.

A	B	C	D	E	F	G	H
1	Položka	Umístění	Velikost	Váha	Presun	Kondice	Hlučnost
2	varchar(255)	int	int	int	int	int	int
3	Pachatel 1	1	2	9	1	9	10
4	Pachatel 2	3	1	10	1	5	7
5	Neutrální osoba 1	7	6	5	10	5	8
6	Neutrální osoba 2	6	3	8	6	6	3
7	Pracovník PKB 1	6	7	2	1	7	4
8	Pracovník PKB 2	7	10	2	8	1	6
9	Pes 1	7	7	2	1	5	5
10	Automobil 1	10	8	1	1	5	6
11	Dodávkový automobil 1	7	3	4	5	1	6
12	Sloup 1	8	5	2	1	0	0
13	Sloup 2	2	10	1	5	0	0
14	Sloup 3	7	5	9	10	0	0
15	PIR 1	2	9	8	9	1	0
16	PIR 2	8	7	2	10	6	0
17	PIR 3	3	3	10	10	9	0
18	Kamera 1	10	1	8	6	10	7

Fig. 4. Sheet Vše (All) of the tool X2_scenar (X2_scenario).

The value of individual attributes is then dependent on assigning these parameters in the sheet Výchozí (Default) as follows:

- For the value “x” (not assigned) in the sheet Výchozí (Default), the value is automatically set to 0, see Fig. 4 (subsequently interpreted as NULL).
- For the value “o” (assigned) in the sheet Výchozí (Default), a random value is set, see Fig. 4.

Random values are provided for each cell using the integrated RANDBETWEEN function as follows:

$$= \text{RANDBETWEEN}(1;10).$$

This default setting ensures the variability of the scenario, for example, the attribute Location acquire values 1 to 10; this can mean different starting points for perpetrators if this attribute is assigned to them. As a result, the scenario is diverse and the positive effect of training on the trainee is increased.

In some cases, the intent of training can be setting specific attribute values for selected objects. These values are fixed by simply inputting them into the appropriate field, which is reflected by the digit being coloured red, see Fig. 4. [7, 8]

The cells of the last sheet Vše (All) contain the already mentioned, automatically generated scripts, see Fig. 5.

These are in particular:

- DROP script for deleting the existing scenario.
- CREATE script for its re-creation.
- INSERT scripts for loading the codebooks with data.

Final merging script, which consists of previous scripts.

Fig. 5. Automatically generated merged script of the tool X2_scenar (X2_scenario).

5 Validation of outputs using the tool for inserting the scenario

The resulting script obtained by means of the tool is then copied and pasted into the MySQL environment. [3] Consequently, it needs to be slightly adjusted. In particular, the following has to be done:

1. Quotation marks need to be deleted at the beginning and at the end of the script.
2. A “comma” at the end of the last line of the script needs to be deleted.

The script modified like this can then be run successfully, as shown in Fig. 6. [7, 8]

Fig. 6. Successful launch of the generated script.

A table shown in Fig. 7 is generated by running this script using a simple command:

*SELECT * FROM tb_scenar_test_1;*

ID	Name	Unit	Value	Priority	Status	Description
1	Prostředí 1	3	10	1	10	3
2	Prostředí 2	3	1	5	2	5
3	Prostředí 3	3	10	5	4	5
4	Prostředí 4	3	10	5	4	5
5	Prostředí 5	3	10	5	4	5
6	Prostředí 6	3	10	5	4	5
7	Prostředí 7	3	10	5	4	5
8	Prostředí 8	3	10	5	4	5
9	Prostředí 9	3	10	5	4	5
10	Prostředí 10	3	10	5	4	5
11	Prostředí 11	3	10	5	4	5
12	Prostředí 12	3	10	5	4	5
13	Prostředí 13	3	10	5	4	5
14	Prostředí 14	3	10	5	4	5
15	Prostředí 15	3	10	5	4	5
16	Prostředí 16	3	10	5	4	5

Time	Action	Message	Duration / Batch
1 01:56:31	CREATE TABLE IF EXISTS scenario_test_1	0 rows affected	0.125 sec
2 01:56:31	CREATE TABLE IF EXISTS scenario_test_1	0 rows affected	0.125 sec
3 01:56:31	INSERT INTO scenario_test_1 (Name, Unit, Value, Priority, Status, Description)	16 rows affected Rows: 16 Duration: 0.076 sec	0.076 sec
4 01:56:31	SELECT * FROM scenario_test_1 LIMIT 5, 1000	16 rows returned	0.000 sec / 0.000 sec

Fig. 7. Scenario table created by running the automatically generated script.

6 Conclusion

This article introduced the design of a tool for facilitating the insertion of new scenarios into an existing virtual simulator. The tool was designed in the MS Excel environment and it enabled generating SQL scripts that directly affected the database of the virtual simulator. These scripts were then run by means of the existing database and they were validated by successfully loading the database with data. Further research related to these issues will concentrate on facilitating the implementation of scenarios and actions into the existing virtual simulator.

This paper is supported by the Internal Grant Agency at Tomas Bata University in Zlín, projects No. IGA/FLKŘ/2019/002 and No IGA/FLKŘ/2018/001.

This research was supported by support research program of Tomas Bata University in Zlín (RVO).

7 References

1. P. Svoboda, *The Use of Simulation in Education of Security Technologies, Systems and Management* (ACMSE 2014, Varna, 2014)
2. P. Svoboda, L. Lukáš, J. Rak, D. Vičar, *The Virtual Training of Hazardous Substances Transportation* (Kaunas University of Technology, Kaunas, 2015)
3. P. Svoboda, Návrh algoritmu implementace virtuálních simulátorů do výcviku v průmyslu komerční bezpečnost (UTB, Zlín, 2018)
4. I. Sommerville, *Software engineering* (Pearson, Boston, 2016)
5. ISO/IEC/IEEE 29148:2011(E), *Systems and software engineering — Life cycle processes — Requirements engineering* (ISO/IEC/IEEE, Piscataway, 2011)
6. R. Pressmann, B. Maxim, *Software engineering: a practitioner's approach* (McGraw-Hill, New York, 2015)
7. P. Svoboda, J. Rak, D. Vičar, M. Zelená, *The Basic Process of Implementing Virtual Simulators into the*

Private Security Industry, (APSAC 2018, Dubrovnik, 2018)

8. P. Svoboda, J. Rak, D. Vičar, M. Zelená, *The Basic Process for the Implementation of Virtual Simulators into the Private Security Industry Using a Support Tool for Automated SQL Script Generation* (WSEAS, Sofia, 2018)