A Proof-of-Concept of a Game Engine-based Simulator for Human Reliability Analysis (HRA)

Marcos Vinicius P. de Andrade, Márcio das Chagas Moura, Isis Didier Lins, Enrique L. Droguett, Marilia A. Ramos.

Marcos Vinicius P. de Andrade, Márcio das Chagas Moura, Isis Didier Lins
Production Engineering Department, Federal University of Pernambuco, Recife, Brazil.

Marilia A. Ramos
Chemical Engineering Department, Federal University of Pernambuco, Recife, Brazil.

Enrique L. Droguett
Mechanical Engineering Department, University of Chile, Santiago, Chile

1. INTRODUCTION

Human behavior and performance have become a major concern in almost every economic activity these days. Today's notion, of human factors as one of the main causes of accidents, is being confirmed by studies in areas such as aviation [1], chemical industry [2], nuclear power generation [3] and water supply [4], among others, turning what was intuitive into a scientific "de facto".

The above-mentioned studies show how important is the predictability of human performance levels and error probabilities in socio-technical systems. However, the scarcity of available data has been an obstacle as many authors attest [5]–[7]. As it can be observed, the lack of human error data continues to be an undesirable presence in the Human Reliability Assessment (HRA) field. This work is motivated by two well-known facts: the lack of good datasets for the HRA estimations, and the shortage of funding for scientific research that generates smaller teams with greater concentration on the end activities leaving few seats to bring in researchers from outside the main area of study.

In this context, recent works bring exciting examples of a possible workaround for this apparently everlasting problem. For example, Refs. [8], [9] present digitally created virtual environments to simulate an evacuation scenario with the resulting data being used to successfully feed a Bayesian Network. Then, the resulting data is used to calculate the conditional probabilities and the likelihood of success for the tasks in study under some predefined circumstances. Another experiment [10] uses driving simulator to study sleepiness at the steering wheel showing also good prospect in the use of a digital double to avoid risky situations as also happens in [11].

Therefore, it is evident the relevance of researching alternatives for systematic data collection beyond traditional post-training assessments, and research or accident reports [12]. Then, the present work proposes to discuss the use of Game Engines (GE) in the construction of virtual environments for Human Reliability Assessment (HRA) studies.

2. QUICK THEORETICAL PARENTHESIS: WHAT IS A GAME ENGINE (GE)?

Game engines consists in a set of game design tools grouped into a unique computational environment., Roughly speaking, GE could be compared to text editors, where all the necessary tools are implemented in the software environment, e.g., printing modules, spelling and formatting. In game engines, in an analogous way, the tools needed from the conception to the final output of an application are implemented and are project-independent.

The various embedded features allow the creation of electronic games, simulations, or any application that requires real-time graphics, simulations and user interaction. The newer GE have layers of hardware abstraction that allow you to create, on a single platform, applications for all other devices. For example, you create on computer desktop games that will run on consoles, mobile platforms like phones and tablets, and on multiple operating systems. The most common features found in modern engines are [13], [14]:

- Graphic engine to generate two-dimensional, three-dimensional and stereoscopic graphics;
- Physics engine for simulations;
- Artificial intelligence (AI) engine for character behaviors;
- Interface for programming languages and script like C ++, C#, JavaScript and others;
- Multi-player network management;
- Virtualizers to simulate the various delivery platforms for prototyping.

From the solutions on the market (or about to be) the following GE are noteworthy:

- Unity Engine www.unity3d.com [15], which is one of the market leaders. It has C# and JavaScript as native programming languages, and has one of the largest user bases on market. However, it has some features missing on its free version;
- Cry Engine www.cryengine.com [16], which boasts one of the best graphic quality on the market and is also an emerging marketing leader with full featured free version and a visual programming language called Flow;
- Ogre www.ogre3d.org is free and open source, has good graphics quality, but has a complex programming interface;
- The giant Amazon is launching Lumberyard aws.amazon.com/pt/lumberyard/ its own proprietary engine for game creation. The applications generated are intended to be distributed on amazon e-commerce platform, it is free and it is in beta phase at the time we write;
- Autodesk, another giant in computer graphics, has also its own engine called Stingray www.autodesk.com/products/stingray/overview, which main features are unprecedent interoperability with many products of its line, like 3ds Max, Maya and Revit. It has a free 3-year license for students.

In the present article, we use the Unreal Game Engine www.unrealengine.com [17], which is free for academic purposes, and if the generated application is sold, Epic Games charges a 5% royalty based on gross revenue for the use of Unreal Engine 4, under the free license agreement, making it a very accessible solution for a kickstart. The software has no working limitations and all versions are available for download at no charge.

The Unreal, among the main features, has a visual programming interface called Blueprints, which permits almost every aspect of the environment to be programmed and controlled without any written code. This feature is especially useful in the perspective of small engineering research teams. This GE has a huge user base and excellent interoperability with the major 3D software packages via FBX file pipeline. Figure 1 shows an example of the created scenarios inside the Unreal GE with an example of its corresponding Blueprint script.

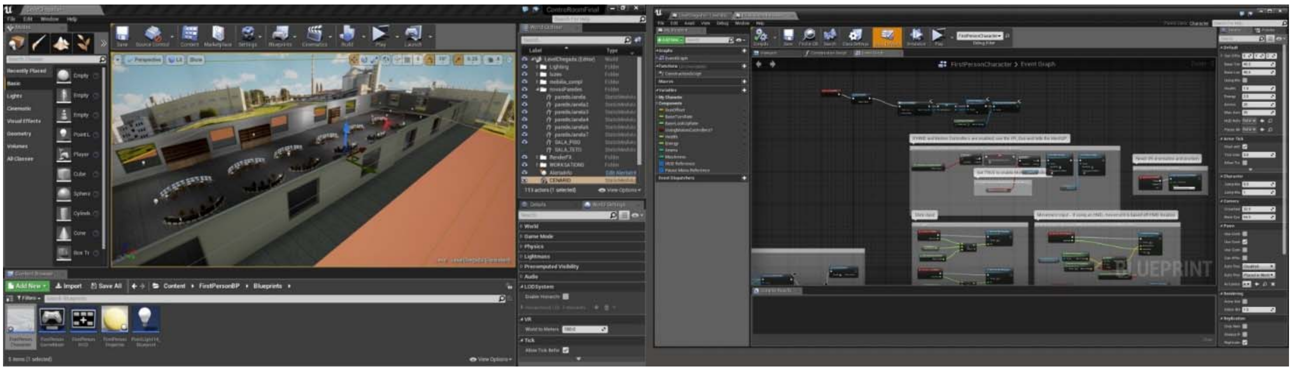

Figure 1 -Scenario overview in the Unreal Game Engine. The blueprint contains environment's programmed behaviors for the simulation. Source: the authors.

3. OBJECTIVES

The present work aims to explore how to create scenarios in a GE-based simulator so that they can be used to feed a human reliability analysis. The following works bring very few, or no info at all, on this subject [8], [18], [19]. At the best of authors' knowledge, Ref [20] is the only one that is a bit clearer on the subject of environment creation though it is not tailored for HRA.

In this way, the present work aims to use GE-based simulations for generating data that can be fed into HRA models. The techniques here discussed are aimed mainly for small research teams. As the team experience grows in building scenarios, the complexity of the implemented solutions can also be extended, but our objective is to start with a simple scenario as literature recommends [21].

4. METHODOLOGY

In this paper, we propose a systematic approach for generating GE based simulator scenarios. For the present work we assume that the data generated will feed a Bayesian Network (BN), but the network creation itself is not within the scope of this work. The references in this work bring a large list of works addressing the subject of BN creation [7], [8], [22]–[24].

Our example consists of an adapted virtual environment based on a scenario described by the event tree shown in Figure 2. In this scenario, as described by authors: *"A small hole on the reformer tubes would leak process gas into the radiation chamber of the reformer. The content of the process gas, especially the hydrogen, may react with the oxygen still present at the combustion gases, which is a very exothermic reaction. The heat produced would increase the temperature of the combustion gases, which, in turn, would heat even more the feed going through the heat exchangers P-01 and P-02 and the combustion air going through P-03. The temperature indicators TI-362, TI-361 and TI-388 would therefore indicate higher temperatures than normal process temperatures, which would be visible to the operator, and the associated High Temperature Alarms (HTA) would sound. …The scenario established in this paper considers the failure of the automatic trip of the reformer. The operator would have then to understand the cues and trip the reformer manually. It also considers that the HTA of TI-361/362/388 will function.…In case the operator does not trip the reformer, the heat generated by the exothermic reactions could increase the temperature above the design temperature of the reformer tubes. This would lead to a catastrophic rupture of the tubes, and a high amount of process gas would rapidly leak into the radiation chamber, which would cause an explosion."* [28]

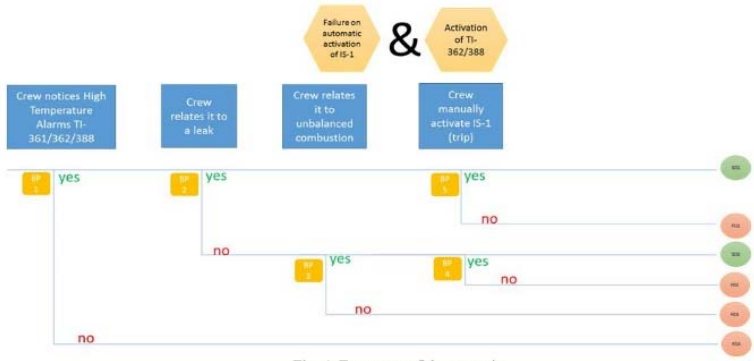

Fig. 4. Event tree of the scenario

The possible outcomes for the scenario are:

S01: Crew notices HTA, relates it to the right cause and trips the reformer
F01: Risk of explosion: crew notices HTA, relates it to the right cause but fails to trip the reformer
S02: Crew notices HTA, relates it to unbalanced combustion, trips the reformer

Figure 2: Event for a scenario that originated the virtual environment. Source: (RAMOS; DROGUETT; MOSLEH; 2016)

For doing such adaptations is proposed here a list of good practices for the proposed approach with a brief description of each step when needed [21], [25], [26]. The final step, which is not conducted in the present work, but it was kept here for sake of illustration of the complete suggested methodology. Then, the steps are:

- Scenario Definition – In this step, all the events to happen in the scenario are physically and logically described with all branches of possible actions covered (Figure 3);
- Choice of the immersion style: will it be Virtual or Augmented Reality or a mix of both?
- Interactions metaphors definition: how does the subject interact in the scenario? – a realistic decision must be made taking in consideration the capabilities of the team. Sometimes, a checkbox with some options may replace complex sequences of animation with very few disadvantages;
- 3D models building, and/or CAD models conversion – Most of the design in the engineering process today is done in 3D. Then, the possibilities of repurposing this already created geometry are huge;
- It is important to have in mind that some adaptations and cleanup in the 3D meshes will be necessary. There are good options of free 3D software like Blender www.blender.org and freeware models to populate the simulation that can be used with minimal adaptations and imported into game engines;
- Objects/Actors placement on scenario: After all the so-called assets are created, they must be "physically" placed on the virtual environment. Parked cars, trees and other elements must be positioned in the virtual environment;
- Intra-game programming of the working laws in the scenarios - Game engines have programming languages to enable environment reactions, logic processing, simulations tuning and any kind of customization needed;
- Event definition and work variables – What will happen as the user/player progresses through the

simulation, how will the relevant data be stored? See Figure 3 and Table 1;
- Quantitative approach definition: what will method be used for HRA? In this article, we considered the used scenario is part of an HRA performed by the Phoenix technique [27];
- Definition of the metrics to be collected during gameplay: In this step all data that will be generated during the game session is defined, e.g. reaction times, actions taken and any other event that is of interest for the study (Table 1);
- Database taxonomy definition: how will the database records be composed of, what will context variables be stored or any other definition regarding data recording must be previously defined;
- Game mechanics policy definition (how much will interference be applied, what will simplifications be done (Figure 3)?). Sometimes some adaptations or some stimulus may be applied intra virtual environment to achieve determined ends, e.g., will be any time constraints? Will user results be visible to other users? Will some manipulations of components be simplified? There will be a right or wrong feedback in real-time for the user? Should all these modifications on how the environment relates to the user be defined in advance?
- Test Gameplay Sessions: Users will interact with the environment to generate data from their actions inside the game;
- Post session debriefing for user impressions (recorded on video or audio only when possible): Is good to listen to the users immediately after a simulated session to catch impressions and valuable insights about what happened inside de virtual scenario. In this step, some flaws can be pointed out permitting to improve the game logic, or to understand how the events where interpreted by the user;
- Benchmark with prior works when possible: Compare with pre-existent data about the studied events, if available.

The virtual environment was based in this adapted scenario to keep the study tractable. The plot defined for the adapted virtual scenario was the crew manager being called to a seat in the control room, where various overheat alarms are on. Then, the manager is asked to decide what to do in this situation via a set of menu options. The environment provided a magnified version of the control monitor that appeared as the user approached a delimited control area showing the plant schematics with alarms on. It's a view very similar to what the operator for the control console sees in the monitor.

In the scenario, all decisions taken by the user are logged, besides that the time spent reading the magnified screen and if the magnification was activated or not. All these features were logged for conditional probability calculation. The event of not shutting down the system on the 2' timeframe will be considered a fail, as also will be a wrong diagnosis at the final debriefing.

In this case, is assumed, as the main purpose, generate data for a BN estimate conditional probabilities of correct scenario diagnostic and decision on tripping the system based on the use of the magnified screen and also in the amount of time spent reading it. Yet, data will be gathered on how much time is spent on reading the alarms screen in comparison with a predefined benchmarked time.

Variables states will be recorded as data flags for reading or not the magnified screen, and the timecodes for activating/deactivating the screen will be saved. This setup will make possible to test conditional probabilities of success on diagnostic and decision as in [8], and their relationship with:

- Human System Interface (HIS) represented by the magnified screen;
- time spent reading the alerts which will be assumed to represent attention PIF.

The event of activating the magnified screen is recorded automatically to a log file allowing the event to be tracked. Yet, a timecode of triggering on and off the magnified info screen will be recorded. All the adaptations and assumptions were summarized at a schematic infographic in Figure 3. The lack of action after the 2 minutes timeframe was linked to severe consequences of the scenario, that would be the rupture of tubes, with a large amount of gas leaking into radiation chamber leading to an explosion [28]. The 2 minutes time limit was defined with no process dynamics in mind, just to give the scenario an acceptable time to finish.

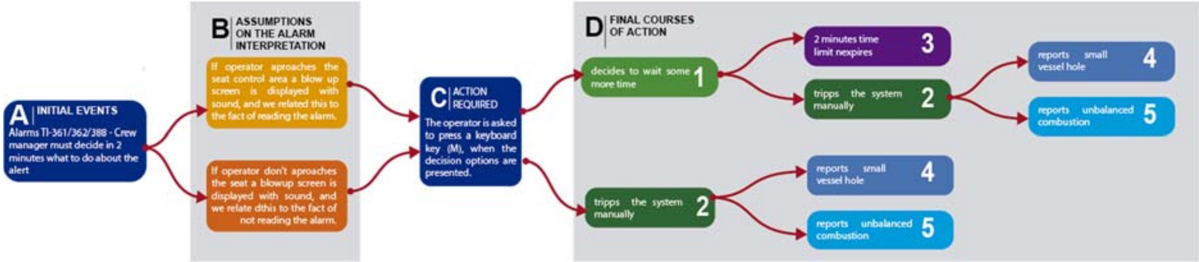

Figure 3: Adapted scenario with assumptions, unfoldings and defined outcomes. Source: authors.

It is necessary to pay special attention to the taxonomy of the database to be inputted so that the generated data can be useful afterwards. There are plenty recommendations in works like [29] and [30], but mainly one must care about characterizing the events variables and its possible states or values, and also taking care to make the collected data compatible with the data base already available for the chosen event.

Table 1 brings a summarized overview of the used variables, which could be extended to store context scenarios, such as amount lighting present in the virtual environment as in [8] and so on. The record for the event will store all the states of the variables throughout the session permitting a description of the event and context as clear as necessary for the study.

Variable Name	Storing	Type	Possible Values	Meaning
SensorMoviment	If the operator activates magnified screen	binary	0, 1	0 - didn´t activate, 1 - otherwise
expiredTimer	if no decision is taken in the 2´ timeframe	binary	0, 1	0 - no decision on timeframe, 1 - otherwise
eventInterpret	the operator interpretation of event	binary	0, 1	0 - small vessel hole, 1 - unbalanced combust.
timeCodeMin	universal time on the scenario minutes	integer	0, 1, 2	the minutes in descending order
timeCodeSec	universal time on the scenario seconds	integer	0....60	the seconds in descending order
timeCodeScreenOn	time stamp of magnified screen activation	integer	0...120	when the user begins to read the screen
timecodeScreenOff	time stamp of magnified screen deactivation	integer	0...120	when the user ceases to read the screen
timeSpentReading	time reading compared to benchmarkd time	binary	0, 1	0 - shorter , 1 - longer

Table 1: Basic set of variables created for the recorded data on the scenario. Source: the authors.

The simulation was constructed partially inside the GE with the other part being modeled in Autodesk 3ds Max [31], and then imported inside the GE via the FBX pipeline, which is an interchange format that maintains much of the data like UVW mapping, textures and some animations.

Almost any 3D modeling, which can output meshes in fbx format, can be used to export assets to Unreal, where they are called actors (Figure 4). After being imported into Unreal, the geometrical 3D meshes must be configured for the basic physics behavior, like mass, friction and how it responds to collision.

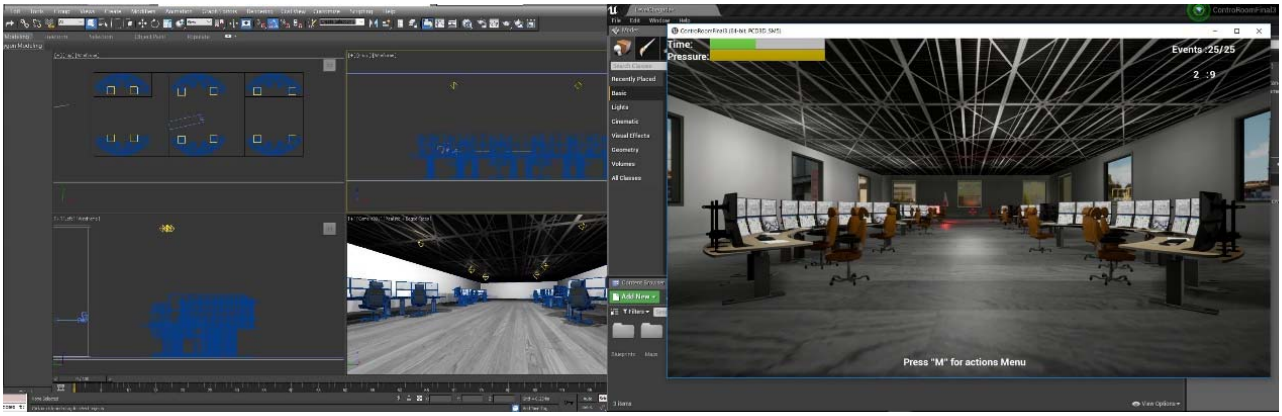

Figure 4 - The modelled scenario inside the 3D software and after imported into Unreal - image by the author.

5. RESULTS AND FINAL IMPRESSIONS

After each session of the simulated virtual scenario, the game engine collects the gameplay programmed variables sates and outputs them to log files, where these states are recorded. Figure 5 shows the Blueprint script implementation of a data gathering event, where the magnified screen activation is recorded, besides the time spent reading it. In a similar way, the time to command a system shutdown, and the options of diagnosis done by the operator are also recorded.

In this proof-of-concept exercise, no dedicated analytics provider was used. Instead of that a write to log file was adopted, which is a debugging tool commonly used for Unreal GE development. If the amount of data grows or the scenario is distributed to multiple locations, a dedicated analytics provider should be used. There is a plethora of solutions, some of them free like GameAnalytics, or extremely low cost like Game DNA that links the generated data straight into Google Analytics.

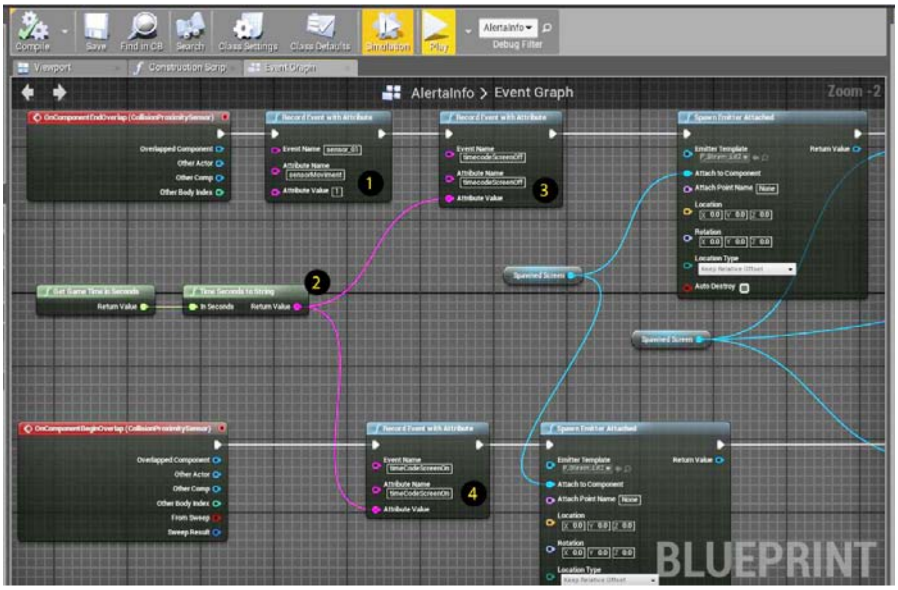

Figure 5 – Blueprint Analytics script:1) logging the activation of magnified screen. 2) generation of the continuous timecode on scenario. Timer variables recording, 3) Timecode of screen activation. 4) Timecode for screen deactivation, permitting to measure the time spent reading it. Source: the authors.

After each session there will be a file with records of all the variables, if the system was shut down on time or not, if the operator correctly diagnosed the event, if the magnified screen was active and how much time it stayed on screen. Then, setting a Bayes Network is a matter of grouping the events and consolidating the obtained data. Studies may be made checking the connection between success in shutting down the system and the amount of time reading the console screen. In a similar way, the precision on event diagnosis may be checked against the fact of reading the alarms on a bigger screen with the event isolated from other alarms to verify interface design issues. The test left a very positive impression, with the application generated by the GE being able to record logs of specific events and output a file with the results as seen in Figure 6. The file format is easy to interpret due to be an ASCII (.txt), which can be easily processed.


d479536f442d4395a9208c9b830380ee-2017.10.04-12.25.44.analytics - Bloco de notas

Arquivo Editar Formatar Exibir Ajuda


```
{
	"sessionId" : "d479536f442d4395a9208c9b830380ee-2017.10.04-12.25.44",
	"userId" : "d479536f442d4395a9208c9b830380ee",
	"events" : [
		{
			"eventName" : "timeCodeScreenOn"
,			"attributes" : [
			{
				"name" : "timeCodeScreenOn",
				"value" : "00:06.96"
			}
			]
		}
,
		{
			"eventName" : "sensor_01"
,			"attributes" : [
			{
				"name" : "sensorMoviment",
				"value" : "1"
			}
			]
		}
,
		{
			"eventName" : "timecodeScreenOff"
,			"attributes" : [
			{
				"name" : "timecodeScreenOff",
				"value" : "00:12.38"
			}
			]
```


Figure 6: Log file generated with records of screen activation and timecodes for calculating reading time. Source: the authors.

This GE tool is free for academic and research use, and the scenarios boast an impressive visual quality and the programming interface is very intuitive and friendly even to non-coders. The overall impression is very promising, and the application example shows that is quite feasible for a team of engineers with some knowledge in 3D CAD to create a scenario to collect data, and then output automatically collected analytics data for HRA studies using GE.

6. LIMITATIONS OF THE WORK

The assumptions in the digital scenario were defined *a priori* with no fine tuning. As in a qualitative phase of an HRA the whole logic of the simulation must be scrutinized to ensure useful data in the end.

Better modeling and animated operators on the workstations would be a plus, but will leave the scenario heavier to process. In a “real-world” situation, they will be mandatory to improve the sensation of immersion for the users, keeping the distance virtual-real as shorter as possible.

The sessions were conducted only with research team members, not involving real operators due to permissions issues. The actual building of the BN was considered out of scope in this paper since it’s been widely addressed in prior works like [7], [8], [22]–[24], so it’s not presented here.

7. FUTURE WORKS

The technology is at a robust state, so testing it in a scenario with available database records will be a much desirable task. Improving the overall graphic quality for the scenarios and animations is also a goal. It’s also intended a test with an analytics provider, so data will be compiled, via web connection, from multiple deployments of the same scenario in different locations with graphic visualizations of data behavior being available.

This developments points out another point to be explored: custom tools to seamlessly integrate generated/compiled data straight into BNs. The task may involve some higher programming skills, but the rewards seem worth the challenge.

References

[1] B. . Kanki, R. . Helmreich, and J. . Anca, *Crew Resource Management*. 2010.

[2] S. G. Kariuki and K. Löwe, “Integrating human factors into process hazard analysis,” *Reliab. Eng. Syst. Saf.*, vol. 92, no. 12, pp. 1764–1773, 2007.

[3] J. Reason, *Human Error*. Nova Iorque, 1990.

[4] S. Wu, S. Hrudey, S. French, T. Bedford, E. Soane, and S. Pollard, “A role for human reliability analysis (HRA) in preventing drinking water incidents and securing safe drinking water,” *Water Res.*, vol. 43, no. 13, pp. 3227–3238, 2009.

[5] a. Carnino, “Human reliability,” *Nucl. Eng. Des.*, vol. 90, no. 3, pp. 365–369, 1985.

[6] L. Zhang, X. He, L. C. Dai, and X. R. Huang, “The simulator experimental study on the operator reliability of Qinshan nuclear power plant,” *Reliab. Eng. Syst. Saf.*, vol. 92, no. 2, pp. 252–259, 2007.

[7] Y. F. Wang, Y. L. Li, B. Zhang, P. N. Yan, and L. Zhang, “Quantitative Risk Analysis of Offshore Fire and Explosion Based on the Analysis of Human and Organizational Factors,” *Math. Probl. Eng.*, vol. 2015, 2015.

[8] M. Musharraf, D. Bradbury-Squires, F. Khan, B. Veitch, S. MacKinnon, and S. Imtiaz, “A virtual experimental technique for data collection for a Bayesian network approach to human reliability analysis,” *Reliab. Eng. Syst. Saf.*, vol. 132, pp. 1–8, Dec. 2014.

[9] M. Musharraf, J. Smith, F. Khan, B. Veitch, and S. MacKinnon, “Incorporating individual differences in human reliability analysis: An extension to the virtual experimental technique,” *Saf. Sci.*, no. November 2016, 2017.

[10] D. Davenne *et al.*, “Reliability of simulator driving tool for evaluation of sleepiness, fatigue and driving performance,” *Accid. Anal. Prev.*, vol. 45, pp. 677–682, 2012.

[11] C. Zhao, M. Zhao, J. Liu, and C. Zheng, “Electroencephalogram and electrocardiograph

assessment of mental fatigue in a driving simulator," *Accid. Anal. Prev.*, vol. 45, pp. 83–90, 2012.
[12] K. M. Groth and A. Mosleh, "A data-informed PIF hierarchy for model-based human reliability analysis," *Reliab. Eng. Syst. Saf.*, vol. 108, pp. 154–174, 2012.
[13] L. Bishop, D. Eberly, T. Whitted, M. Finch, and M. Shantz, "Designing a PC game engine," *IEEE Comput. Graph. Appl.*, vol. 18, no. 1, pp. 46–53, 1998.
[14] R. Darken, P. McDowell, and E. Johnson, "Projects in VR: The Delta3D open source game engine," *IEEE Comput. Graph. Appl.*, vol. 25, no. 3, pp. 10–12, 2005.
[15] Unity, "Unity Inc.," *2015*, 2015. [Online]. Available: unity3d.com.
[16] Cry Engine, "Cry Engine." [Online]. Available: https://www.cryengine.com.
[17] I. Epic Games, "Unreal Engine," *Unrealengine.Com*, 2017. [Online]. Available: https://www.unrealengine.com/what-is-unreal-engine-4.
[18] Y. Ba, W. Zhang, G. Salvendy, A. S. K. Cheng, and P. Ventsislavova, "Assessments of risky driving: A Go/No-Go simulator driving task to evaluate risky decision-making and associated behavioral patterns," *Appl. Ergon.*, vol. 52, pp. 265–274, Jan. 2016.
[19] D. Cohen *et al.*, "Emergency preparedness in the 21st century: Training and preparation modules in virtual environments," *Resuscitation*, vol. 84, no. 1, pp. 78–84, 2013.
[20] D. Cohen *et al.*, "Tactical and operational response to major incidents: Feasibility and reliability of skills assessment using novel virtual environments," *Resuscitation*, vol. 84, no. 7, pp. 992–998, 2013.
[21] E. Adams, *Fundamentals of Game Design*, 2nd ed. Berkeley, CA: New Riders, 2013.
[22] K. M. Groth and L. P. Swiler, "Bridging the gap between HRA research and HRA practice: A Bayesian network version of SPAR-H," *Reliab. Eng. Syst. Saf.*, vol. 115, pp. 33–42, 2013.
[23] K. M. Groth, C. L. Smith, and L. P. Swiler, "A Bayesian method for using simulator data to enhance human error probabilities assigned by existing HRA methods," *Reliab. Eng. Syst. Saf.*, vol. 128, pp. 32–40, 2014.
[24] M. Musharraf, J. Hassan, F. Khan, B. Veitch, S. MacKinnon, and S. Imtiaz, "Human reliability assessment during offshore emergency conditions," *Saf. Sci.*, vol. 59, pp. 19–27, 2013.
[25] S. Brambilla and D. Manca, "Recommended features of an industrial accident simulator for the training of operators," *J. Loss Prev. Process Ind.*, vol. 24, no. 4, pp. 344–355, 2011.
[26] D. Manca, S. Brambilla, and S. Colombo, "Bridging between Virtual Reality and accident simulation for training of process-industry operators," *Adv. Eng. Softw.*, vol. 55, pp. 1–9, 2013.
[27] N. J. Ekanem, A. Mosleh, and S.-H. Shen, "Phoenix–A model-based Human reliability analysis methodology: Qualitative analysis procedure," *Reliab. Eng. Syst. Saf.*, vol. 145, pp. 301–315, 2015.
[28] M. A. Ramos, E. L. Droguett, and A. Mosleh, "Human Reliability Analysis of an Oil Refinery Operation Using Phoenix Methodology: a Hydrogen Generation Unit Case Study," vol. 7, no. 4, pp. 1633–1639, 2016.
[29] J. Park *et al.*, "A Guideline to HRA Data Collection from Simulations," *Int. J. Performability Eng.*, vol. 10, no. 7, pp. 729–740, 2014.
[30] Y. Kim, J. Park, and W. Jung, "A classification scheme of erroneous behaviors for human error probability estimations based on simulator data," *Reliab. Eng. Syst. Saf.*, vol. 163, no. February, pp. 1–13, 2017.
[31] Autodesk, "Autodesk," 2017. [Online]. Available: www.autodesk.com.