Automated Scenario Generation for Training of Humanitarian Responders in High-Risk Settings

Ahmed A. Abdelgawad, Nadia Noori, Tina Comes

Department of ICT
University of Agder
Grimstad, Norway
{ahmedg/nadia.noori/tina.comes}@uia.no

***Abstract* — Scenarios are designed to support decision-makers in gaining a better understanding of the consequences of their decisions. Scenario-based simulations, hence, present an ideal way to train decision-makers for complex and high-risk situations. Usually, training scenarios are human-authored which is limiting in terms of the number and diversity of the generated scenarios, in addition to being a time- and resource-consuming process. In this paper, we are introducing a first-order logic rule-based design and implementation of an automated scenario generator system to support designers of virtual, augmented and simulated reality humanitarian training setups. The system is based on knowledge extracted from publicly available humanitarian related databases. Moreover, the system can produce scenarios that adapt to user performance in addition to meeting certain training requirements.**

Keywords - Automated Scenario Generation; Humanitarians Training; First-order logic; Rule-based system

I. INTRODUCTION

Scenarios are designed to support decision-makers in gaining a better understanding of the consequences of their decisions. Scenario-based simulations hence present an ideal way to prepare decision-makers for complex and high-risk situations [1], [2]. At the same time, due to the complexity of strategic decisions and the resulting multitude of scenarios, there is always a need to select scenarios to fully work out in and choose the appropriate level of detail and granularity [3]. One of the challenges in the design of scenario-based training is thus to find the scenarios that maximise learning while adhering to given levels of complexity [4], reliability and fidelity [5] time or resources for the training [6].

Training is also an ideal environment to test and evaluate new technologies, such as the tracking and monitoring tools that are developed in the iTRACK project. The iTRACK is a European project funded by H2020 – under the Secure Societies Program. The iTRACK project aims at developing an "integrated system for real-time TRACKing and collective intelligence in civilian humanitarian missions" [7], especially in the high-risk zones in the Middle East. The project aims to create a platform that provides tools for improving the protection and safety of humanitarian missions with intelligent socio-technical solutions to support tracking, threat detection, navigation, logistics, and coordination in humanitarian disasters.

Along with the complexity of actors, users, technologies, and cases, also the requirements for scenarios are getting increasingly complex, and traditional exercise planning methods that are rooted in brainstorm sessions and discussions fall short in exploring all potential eventualities. Making use of automated scenarios also facilitates run-time updates of the scenario, based on the performance of players or information on the external environment such as weather data [8]. Automation also offers possibilities of real-time evaluation for rapid feedback loops that have proven to be essential for scenario-based learning [9].

Our goal is to design and implement a scenario generator system to support designers of virtual, augmented and simulated reality environment training missions, in addition to supporting the iTRACK project training and testing sessions; Triplex [10] is another example of a simulated reality environment that could benefit from such a system. Generally, in such training setups, training scenarios are human-authored which is limiting in terms of the numbers of authored scenarios and their diversity, in addition to being a time- and resource-consuming process. In this paper, we are introducing an automated scenario generator system that can produce an unlimited number of training scenarios, which adapt to user (trainee/player) performance in addition to meeting certain training requirements in the field of humanitarian aid provision. The system is based on simple first-order logic rules and a knowledge base that is extracted from publicly available humanitarian aid related databases.

This paper is divided as follows: Section II, the following section, provides a background. Section III includes a description of the scenario generator system. This description covers the system design and implementation. Section IV presents our results in evaluating the system. Finally, Section V provides our concluding remarks.

II. BACKGROUND

Scenario-based methods are used in different fields such as innovation, healthcare, education and others including

disaster and crisis management. The iTRACK target user-groups are operating in environments characterised by being uncertain with high-risk levels. Therefore, the methods proposed in designing the iTRACK scenarios should consider these characteristics.

Scenarios and gaming are widespread in policy analysis and decision support. Scenarios are used in policy decision making to determine the effects of possible decisions on the future. A scenario in this setting represents a possible future. Having a set of different scenarios (and thus different possible futures), uncertainty about the future will be reduced. The scenario-based reasoning makes it easier to prepare for an uncertain future and therefore making a better-informed decision.

Scenario thinking has been described as one way to plan in complex situations and under uncertainty [11]–[13]. In these contexts, scenarios should not be understood as predictions or forecasts. Rather, they are a means to explore potentially relevant future developments before they occur [14]. In our context, scenario thinking is purposeful and targeted at making a specific (strategic) decision.

In this research work, we will use a formalized structure to describe various threat scenarios that make the relevant variables and their interdependencies explicit. This approach will provide an inventory of scenarios that can be the basis for games as well as training simulations, by combining the threat scenarios with other events to "scenario trees" that unfold with the game.

There is a distinction between a scenario and a game in development planning, as they are viewed as two separate entities. Scenarios are seen as being largely model based and/or conceptual representations of the future. Games, on the other hand, allow a more experiential and social interactive exploration of the future [15]. In the iTRACK, we are integrating both approaches into the platform to help forecast and mitigate the risks of operations executed in high-risk and uncertain environments.

For the iTRACK, an important feature to scenario development is the ability to adapt the process of building scenarios to accommodate high levels of uncertainty. Therefore, the framework needs to be modular to mix and match different components during the process of scenario planning and building. Modularity will provide a great degree of freedom in the range of scenarios planned using the scenario generator.

Criteria defined for validating the generated scenarios are required to produce a realistic training environment and improve the ability to forecast the risks involved in humanitarian work in conflict zones [13], [16]. The main guidelines for validating the scenario adequacy are:

- Evaluation of participants' performance [17], [18].
- Evaluation of scenario generator performance.
- Evaluation and assessment of the iTRACK technology, policies and procedures.

Scenario adaptation is a process designed to augment the ability of a single human scenario author to deliver personalised learning experiences to numerous individual learners. The key advancement is to inject automation into the traditional scenario-authoring pipeline. The traditional scenario authoring means that a scenario is manually authored and then played by a learner in a virtual game or simulation environment. A trainer may be present to track the learner's progress in the simulation and record changes to the learner's attributes or performance measurements are collected automatically. In contrast, scenario adaptation is the process having a cycle from simulation to a trainer to scenario adaptation that facilitates a single manually authored scenario to be recycled by progressively adapting to the learner's changing needs and abilities. The scenario adaptation process can be done by "rewriting" manually authored scenarios, by leveraging the human intuition to create a theoretically exponential number of unique experiences. Modifications to the human-authored scenario may take the following forms:

- Adding events and elements to provide additional opportunities to practice a certain skill, or to address an additional learning objective.
- Removing events and elements when the learner is already proficient.
- Changing the details of events to make them easier or harder.

III. SYSTEM DESCRIPTION

A. System Design

A1. Scenario Generation and Data Sources

While building the scenario generator system, the main goal was to make it capable of generating realistic and plausible scenarios as much as possible. Accordingly, this could be used in training the humanitarian aid workers in dealing with most of the security threats and situations they might face in the field. To achieve this goal, the system uses incidents from the publicly available Aid Workers Security Database (AWSD) [19] as the source of security threats and incidents' properties.

AWSD database could be downloaded from the Internet in one table that includes records about humanitarian aid workers security incidents. Each record covers location (e.g. roads/convoy in transit, refugee camps, hospitals, warehouses, offices, etc.), attack context (e.g. ambushes, attacks on road, combats, raids on office or project site, etc.), means of attack/attack weapons (e.g. small arms shooting, bombing, aerial bombardment, etc.), and the number of casualties and the incident description, in addition to some other fields. The AWSD table was imported to our knowledge base under the relation instance $Incident(i, cn, lc, ac, ma, k, w, kk, \dots)$. For simplicity in writing, we will continue using the relation instance $Incident(i, cn, lc, ac, ma, k, w, kk)$ instead, where i is the

incident id, *cn* is the country of incident, *lc* is the location of the incident, *ac* is the attack context, *ma* is the means of attack, *k* is the number of the kidnapped aid workers, *w* is the number of the wounded aid workers, and *kk* is the number of the killed aid workers. The other fields although still exist in our knowledge base are not used. They were kept to the maintain the original schema of the AWSD to make importing new records in the future easy for the scenario generator system administrator(s).

Based on the data extracted from the AWSD, the scenario generator covers sets of training situations. These situations are based on an incident that happened in different locations in the countries the iTRACK project focuses on in the Middle East, namely Iraq, Syria, and Yemen. The scenario generator arranges the AWSD incidents into independent sets based on combinations of similar locations, attack contexts, and means of attack –see Figure 1 and Table 1 for examples of these combinations. The system automatically calculates the frequencies of these sets from the AWSD records connected to the focused countries. Figure 1 shows a cut-out example of a scenario tree [20] based on these calculated frequencies. As an example –see the green dashed path in Figure 1– all incidents happened in the MENA region were 566, of them 267 happened on roads. 225 of these 267 were ambushes (AM). 98 of the 225 ambushes were done using small arms (S). The whole MENA region countries were chosen to be focused on when considering the incidents of AWSD because of the geographical/political similarities with the iTRACK focused countries. In Table 1, the probability of a scenario is calculated as the frequency of the combination of its location, attack context and means of attack divided by total incidents in the MENA region.

In the first generated training mission, the scenario generator provides a scenario from the most frequent branch –see the green dashed path in Figure 1. To generate the subsequent scenarios with difficulty levels that adapt to the performance of the users, the scenario generator receives the user's performance score as an input (depicted in Figure 7). Calculating the user's performance score is not part of the scenario generator, as it is supposed to be calculated by an observer or a trainer either based on preferred performance measures and any other measures related to the training session design and goals. For the scenario generator system, if the user's performance score is higher than a pre-defined threshold, the scenario generator provides a more complex scenario based on the next lower relative frequent branch of the scenario tree (see the orange dotted path in Figure 1). Higher relative frequency (i.e. higher probability) scenarios in the AWSD was taken as a proxy of the easiness of the scenario, as it is more probable that the humanitarian aid workers have faced the more frequent scenario or have been trained before to face it, or at least have heard about it compared to the less frequent scenario.

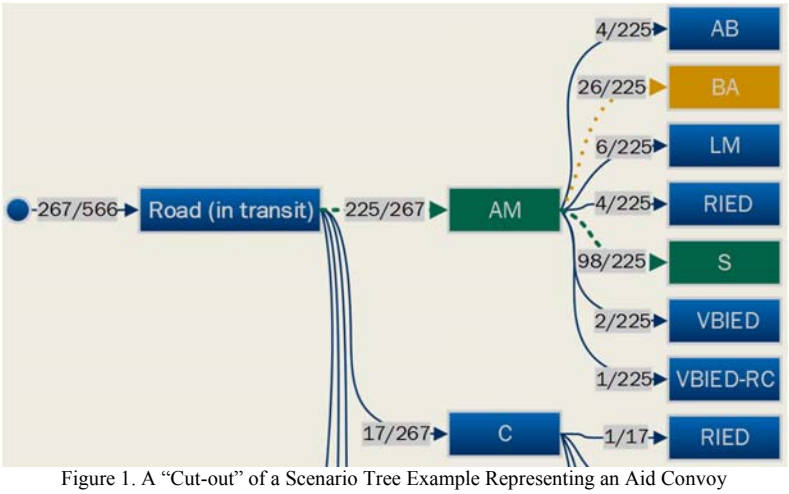

Figure 1. A "Cut-out" of a Scenario Tree Example Representing an Aid Convoy

TABLE 1. EXAMPLES OF SCENARIOS BASED ON POSSIBLE SECURITY INCIDENTS

Location	Attack Context	Means of Attack	Scenario	Probability
R	AM	S	R->AM->S	0.4505
R	AM	BA	R->AM->BA	0.0459
…				
R	C	S	R->C->S	0.0141
…				

Training humanitarian aid workers to deal with security threats as mentioned above is a key demand in the scenario generator, however, security events are not the only threats that the humanitarian aid workers might face during their missions. Accordingly, to generate vertically scalable scenarios, the scenario generator, in addition to security threats, generates natural, mechanical, traffic, technical, iTRACK technology threats based on pre-defined probabilities.

Moreover, not all attackers groups operate in all cities/areas. Accordingly, for a realistic scenario, the scenario generator system provides a training mission with a matching pair of a city and an attackers group based on data extracted from the Global Terrorism Database (GTD) [21]. In case the mission is a convoy "Road (in transit)", the scenario generator supplies a set of two cities with a reasonable pre-defined distance between them (so that they are not very close or too far for a realistic convoy). The GTD is downloadable from the Internet in one table that was imported to our knowledge base under the relation instance $CityAttacker(i, cn, cty, lat, lng, agr, \dots)$.

For simplicity, $CityAttacker(i, cn, cty, lat, lng, agr)$ will be used instead, where *i* is the record id, *cn* is the country of incident, *cty* is the city of incident, *lat* is the latitude of the city, *lng* is the longitude of the city, and *agr* is the attackers group name. To produce a city and an attackers group name for the generated scenario, the scenario generator system uses a query similar to the following:

$$\theta = \exists i, cn.\left(CityAttacker(i, cn, cty, lat, lng, agr) \wedge (cn = 'Syria')\right) \quad (1)$$

A2. Scenario Impact

The scenario generator produces possible impacts for a security incident based on the incident location, context of attack, and means of attack. In our knowledge base, the impacts of the security incidents are stored in the relation instance $IncidentImpact(i, lc, ac, ma, im, pb)$, where *i* is the impact id, *lc* is the location of the incident, *ac* is the attack context, *ma* is the means of attack, *im* is the impact, and *pb* is the probability of this impact happening. Table 2 shows examples of possible impacts, and the locations, the attack contexts and the means of attack they are related to in our knowledge base (note that '*' in our knowledge base is used as a wildcard. For example, if there was no record that contains *lc='R'* in the knowledge base, the record having the wildcard will be used instead). Accordingly, the scenario generator system executes a query similar to the following to produce a suitable security threat impact for the generated scenario:

$$\psi_s = \exists i, lc, ac, ma.\left(IncidentImpact(i, lc, ac, ma, im, pb) \wedge (lc =' R') \wedge (ac =' AM') \wedge (ma =' S')\right) \quad (2)$$

The natural events and their corresponding impacts were added to our knowledge base under the relation instance $NaturalEvent(i, cn, lc, dsc, fd, td, im, pb)$, where *i* is the id, *cn* is the country of event, *lc* is the location, *dsc* is the description of the event, *fd* and *td* are the expected from-date and to-date respectively, im is the expected impact of the event, and *pb* is the probability that this impact will happen. Table 3 shows examples of natural events and their related impacts in our knowledge base.

Not every natural event –or any other event type— must be suitable to happen at any time during the year. Accordingly, the scenario generator accepts a user-defined date of the scenario to be generated (in day and month), otherwise uses today's day and month instead (assumed 26.10 in the following example). The scenario generator system executes a query similar to the following to produce a natural event for the generated scenario:

$$\begin{aligned}\psi_n = \exists i, cn, lc.\Big(&NaturalEvent(i, cn, lc, dsc, fd, td, im, pb) \wedge (c = 'Syria') \wedge (lc = 'R') \\ &\wedge \Big(\big(\leq (fd, '26.10') \wedge \geq (td, '26.10') \wedge \leq (fd, td)\big) \\ &\vee \big(\geq (fd, '26.10') \wedge \leq (td, '26.10') \wedge > (fd, td)\big)\Big)\Big)\end{aligned} \quad (3)$$

Each of ψ_s and ψ_n might normally contain more than one suitable impact. In all cases, the scenario generator will use randomly generated numbers to select one impact from each set based on the value of the probability field connected to that impact record. The same method is followed for producing the impacts of mechanical, traffic, technical, and iTRACK technology threats (sometimes one of these impacts is no impact at all, in such a case the related threat type will not appear in the results).

TABLE 2. EXAMPLES OF SECURITY INCIDENTS IMPACT

Location	Attack context	Means of attack	Impact	Impact probability
…				
R	*	*	Convoy was hit	0.33
R	*	*	Convoy held/detained at checkpoint	0.33
…				
*	*	*	Partly hit	0.5
*	*	*	Hit	0.5
…				

TABLE 3. EXAMPLES OF NATURAL EVENTS AND THEIR RELATED IMPACTS

Country	Location	Description	From date	To date	Probability	Impact
…						
*	*	Snow	01.12	01.03	0.05	Frostbite
*	*	Heat waves	01.06	01.09	0.5	Heatstroke
…						

A3. Damage Calculation

Different damage types and damage amounts are probable because of any of the threats/events facing humanitarian aid workers in the field. AWSD focuses by design on collecting information on the damage happened to the personnel, like being killed (KK), kidnapped (K), wounded (W), and raped or sexually assaulted (RSA). However, covering other types of damages is important to provide more comprehensive training scenarios. AWSD has no information about damage that happened to vehicles (V), buildings (B), equipment (E), or aid commodities (C). To fix this situation, our research team extract these damage types based on the incident description and other fields provided in AWSD records for the MENA region countries. These newly extracted fields were inserted to a new relation instance connected to $Incident$ with a one-to-one relation. $IncidentFixed(i, lc, ac, ma, r, v, b, e, c)$ is this newly added relation instance, where i is the id, *lc, ac*, and *ma* are location, attack context and means of attack (the same like AWSD, yet sometimes manually altered by the administrator(s) to accommodate for details that were mentioned in the AWSD incident textual description), *r* is the RSA, and *v, b, e, c*, are the V, B, E, and C (mentioned above) respectively. Accordingly, the scenario generator system executes a query similar to the following to find the possible damage types for the generated scenario:

$$\varphi = \exists i, cn, lc, ac, ma.\big(Incident(i, cn, k, w, kk) \wedge IncidentFixed(i, lc, ac, ma, r, v, e, c, b) \wedge (cn = 'Syria') \wedge (lc = 'R') \wedge (ac = 'AM') \wedge (ma = 'S')\big) \quad (4)$$

This query decides the possible damage types for the generated scenario, yet the system still needs to generate damage value for each type based on the impact generated earlier. The scenario generator has a pre-defined pair of a min and max probability values that is connected with each impact and based on each damage types– see Table 4. In our knowledge base, these probability pairs connected with damage types are stored in the relation instance $IncidentDamage(i2, dt, pbmn, pbmx, i)$, where *i2* is the id of the record of the damage type, *dt* is the damage type, *pbmn* and *pbmx* are min and max probability values respectively, and *i* is the id of the impact that this damage type is connected to. It should be noted that the damage types resulted from equation 4 supersede the damage types related to impacts in Table 4.

The scenario generator uses these min and max probability values in generating a reasonable random damage amount per type. There are cases where the same damage type happens because of several threat types, for example, a security threat could cause a vehicle damage, also a mechanical threat could cause a vehicle damage. In such a case, the scenario generator will use the min and max probability values that yield the widest range to generate the damage value for this damage type. The scenario generator system executes a query similar to the following to produce the security incident impact and its related damage for the generated scenario:

$$\varphi_s = \exists i, i2, lc, ac, ma, im, pb.\big(IncidentImpact(i, lc, ac, ma, im, pb) \wedge IncidentDamage(i2, dt, pbmn, pbmx, i) \wedge (lc = 'R') \wedge (ac = 'AM') \wedge (ma = 'S')\big) \quad (5)$$

Additionally, in our knowledge base, the damage types connected to the impacts of natural events are stored in the instance $NaturalEventDamage(i2, dt, pbmn, pbmx, i)$, where *i2* is the id of the record of the damage type, *dt* is the damage type, *pbmn* and *pbmx* are min and max probability values respectively, and *i* is the id of the natural event impact that this damage type is connected to. The scenario generator system executes a query similar to the following to produce the natural event impact and its related damage for the generated scenario – see Table 5 which contains damage types and values based on impact related to natural events:

$$\varphi_n = \exists i, i2, lc, dsc, fd, td, im, pb.\big(NaturalEvent(i, cn, lc, dsc, fd, td, im, pb) \wedge NaturalEventDamage(i2, dt, pbmn, pbmx, i) \wedge (cn = 'Syria') \wedge (lc = 'R')\big) \quad (6)$$

For the generated scenario also, the scenario generator system executes queries similar to φ_n, to produce the mechanical, traffic, technical, and iTRACK technology events' impacts –if any of them exists in the generated scenario– and their related damages.

To keep the generated damage values plausible for users, an additional rule was taken into consideration when calculating the combination of damage amounts happening to the personnel (KK, K, W and RSA). Personnel could be kidnapped, wounded, and raped in the same time, however, all these at one side and being killed on the other side are mutually exclusive events (although possible in reality, it would be just confusing to the user). Venn diagram of this rule is shown in Figure 2.

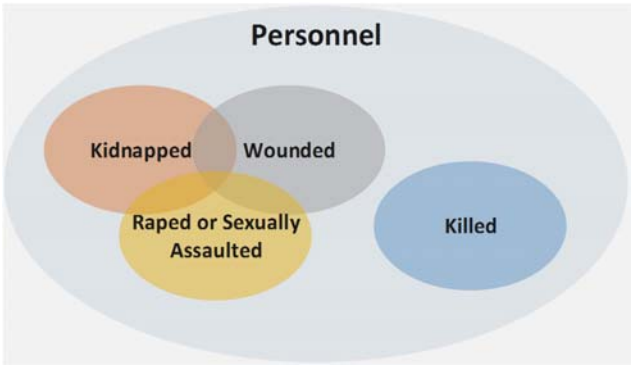

Figure 2. Damage in Personnel

TABLE 4. EXAMPLES OF DAMAGES OF IMPACTS RELATED TO SECURITY INCIDENTS

Damage type	Damage probability min	Damage probability max	Impact id
…			
*	0.2	0.8	10
*	0.8	1	11
…			
KK	0	0	7
E	0.8	1	7
…			

TABLE 5. EXAMPLES OF DAMAGES OF IMPACTS RELATED TO NATURAL EVENTS

Damage type	Damage probability min	Damage probability max	Natural event id
…			
C	0	1	4
E	0	1	4
KK	0	1	5
C	0	1	5
…			

A snapshot of the scenario generator system depicting a scenario block diagram presenting several events/threats with their connected impacts and damage types is shown in Figure 6. Figure 7 shows another snapshot of the system displaying the related damage values as well.

A4. Damage Mitigation

Based on the generated scenario, the scenario generator supplies a set of suitable Standard Operations Procedures (SOPs) and policies in checkbox format as shown in Figure 7. These SOPs and policies were mainly extracted from the United Nations Department of Safety and Security (UNDSS) training material [22], and the European Interagency Security Forum (EISF)'s basic guide for smaller NGOs [23]. To confirm implementing or applying any of these SOPs or policies, the scenario generator exercise observer or trainer can check any of these checkboxes causing a decrease in the amount of damage for certain damage types based on a set of pre-defined rules. Because of the enormous needed amount of data, in calculating these damage mitigation effects, the SOPs and policies damage mitigation effects were considered mutually exclusive.

Finally, to generate horizontally scalable scenarios, or as we call it management view/training, the scenario generator can generate multiple parallel scenarios to train management personnel on managing different teams in different situations.

B. System Implementation

B1. Used Technologies

The scenario generator is Graphical User Interface (GUI) web-based, in addition, supplies a Representational State Transfer Application Programming Interface (REST API) to be able to communicate its generated scenarios to other computer systems including mobile applications. The system was built using common standard web technologies: HyperText Markup Language (HTML), Cascading Style Sheets (CSS), and JavaScript [24] on the client-side. On the server-side, the scenario generator uses Python/Django web-service framework. According to the W3Techs (World Wide Web Technology Surveys), as a programming language for web-services, Python is not popular. Python is used by merely 0.2% of all websites whose server-side programming languages are known to the W3Techs [25]. The rationale behind selecting Python as the programming language for the scenario generator is basically its popularity among data scientists. According to the KDNuggets software poll in 2016, Python came in the second position after R with a share of 45.8%, with +51% growth over 2015 [26]. This could be taken as a proxy for the availability of several Python packages that could help in developing a scientific/data science application like our scenario generator system.

B2. System Data Flow

Figure 3 shows the context diagram or level zero Data Flow Diagram of the scenario generator system. The main external entities in addition to the 'User' are the 'GraphViz'

software, the 'Scenarios Database' (hosting our knowledge base) on MySQL Server, and the 'Static Web Content' on the system's web server. GraphViz is an open source graph visualisation software. It "is a way of representing structural information as diagrams of abstract graphs and networks" [27]. For faster and easier scenario reading, the scenario generator uses GraphViz to draw block diagrams for the scenarios it generates (See an example in Figure 6). The Scenarios database has several tables related to the system management in addition to the main tables which the system uses to generate scenarios (keeping data of the relations instances mentioned above).

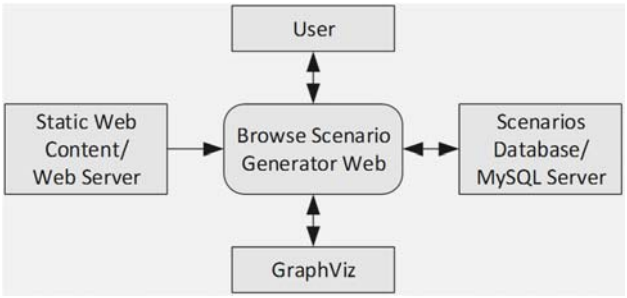

Figure 3. Context Level Data Flow Diagram of the Scenario Generator System

Scenarios and their block diagrams are generated by the scenario generator system based on the User's provided parameters and the data stored in the Scenarios database at one side, and GraphViz from the other side respectively. Accordingly, communication between the main process of browsing the scenario generator Web and other entities are two ways in all cases except with the Static Web Content, taking into consideration that User can edit data in the Scenarios database when she/he conducts system management as an administrator.

B3. System Graphical User Interface

For human readability, the scenario generator system produces a textual description of the generated scenario mission and events as shown in Figure 4. Based on the generated city latitude and longitude, the system shows a map of the mission city (in the case of a convoy, a map for the departure city and another for the destination are shown instead) as shown in Figure 5. For modularity of presenting the scenarios, the system produces a block diagram for the generated scenario as shown in Figure 6. Moreover, based on the generated scenario, the scenario generator supplies a set of suitable Standard Operations Procedures (SOPs) and policies in the form of checkboxes as shown in Figure 7. After supplying the score, another scenario is presented under the current one. Figure 8 presents the management view, showing multiple teams. Each team has an independent thread of scenarios.

The system supplies this admin web-view for the administrator(s) to maintain the knowledge base of the system, in addition to the database tables related to the system management. Figure 9, Figure 10 and Figure 11 show three snapshots of the password protected admin view of the scenario generator system. Figure 9, shows all system database tables including system user management tables. Figure 10 presents an editable view of one security incident extracted from AWSD, it also shows its related damage types. Figure 11 presents an editable view of one natural event, its related damage types, and the min and max probabilities connected with each damage type.

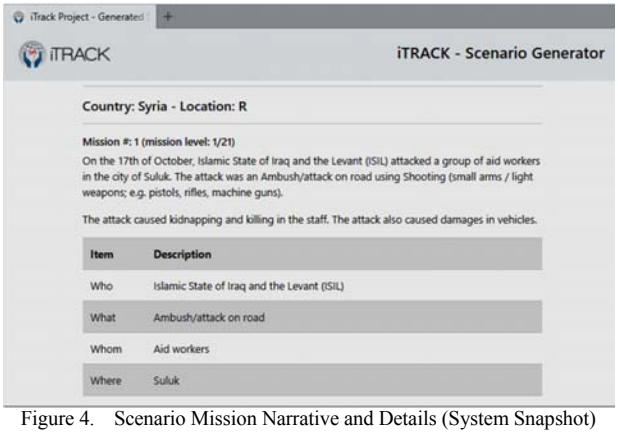

Figure 4. Scenario Mission Narrative and Details (System Snapshot)

Figure 5. Scenario Mission City Map (System Snapshot)

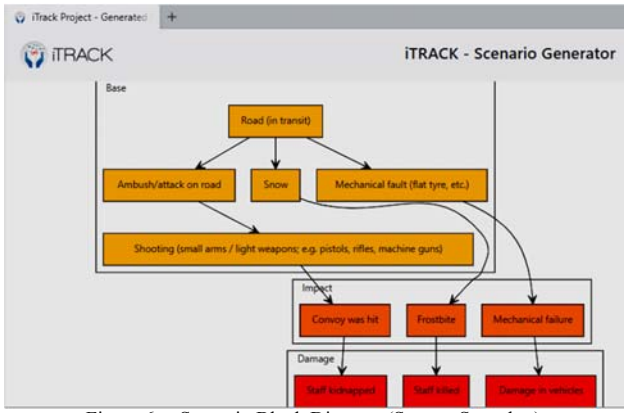

Figure 6. Scenario Block Diagram (System Snapshot)

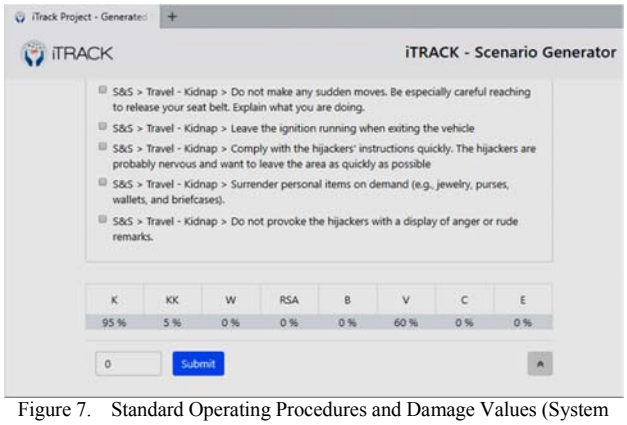

Figure 7. Standard Operating Procedures and Damage Values (System Snapshot)

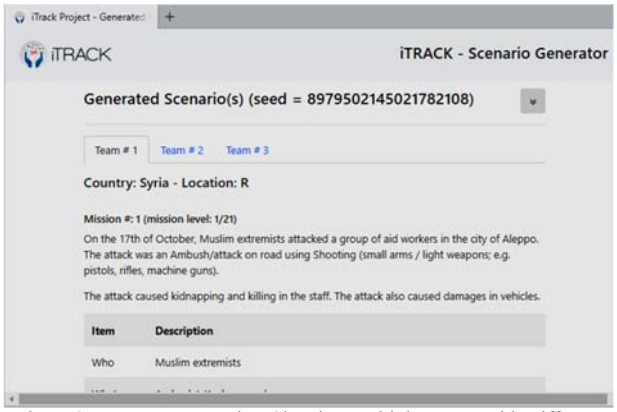

Figure 8. Management View Showing Multiple Teams with Different Scenarios (System Snapshot)

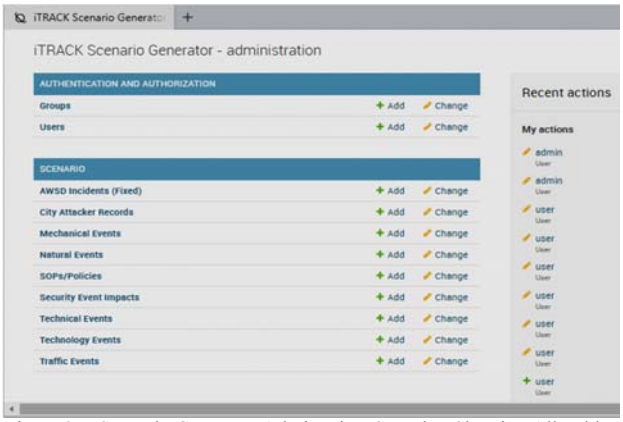

Figure 9. Scenario Generator Admin View Snapshot Showing All Tables Including System Users and Groups

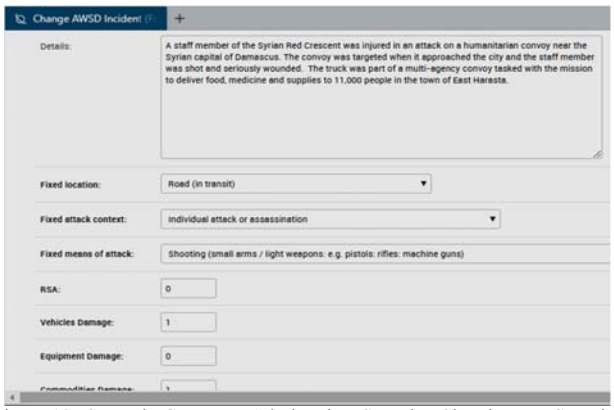

Figure 10. Scenario Generator Admin View Snapshot Showing one Security Incident (Details from AWSD) and its Related Damage Types

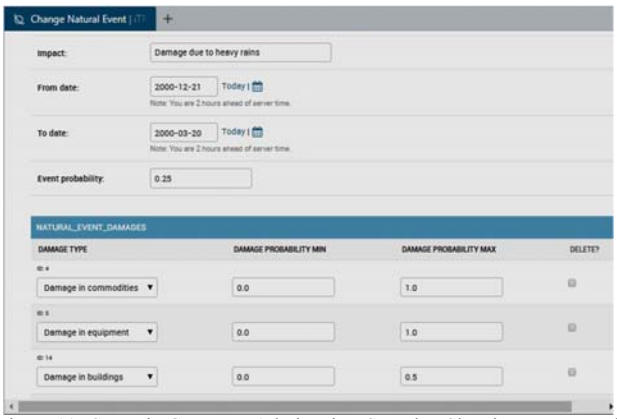

Figure 11. Scenario Generator Admin View Snapshot Showing one Natural Event and its Related Damage Types

IV. SYSTEM EVALUATION

The system has been reviewed by three of the iTRACK project partners that belong to academia, software development, and humanitarian aid organisation, and their notes were taken into consideration in the final version of the system. Furthermore, as suggested in [28], to test the diversity of scenarios generated by the scenario generator, we have converted the scenarios generated by REST API interface to strings and used the Levenshtein distance to measure the distance between the first generated scenario and all successive scenarios. The generated scenarios were shown to be diversified according to the results of this test as depicted in Figure 12.

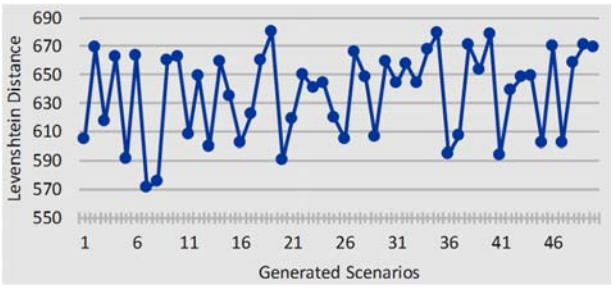

Figure 12. Levenshtein Distance between The 1st Scenario and All Successive Scenarios

V. CONCLUDING REMARKS

This paper presented a design and implementation of an automated scenario generator system that could be used to support designers and trainers of virtual, augmented, and simulated training humanitarian aid missions. Generally, in such training setups, training scenarios are human-authored which is limiting in terms of the number of authored scenarios and their diversity, in addition to being a time- and resource-consuming process. The system capable of producing an unlimited number of training scenarios, which adapt to user performance in addition to meeting certain training requirements. The system is based on simple first-order logic rules that are extracted from publicly available humanitarian related databases. The system has been reviewed by the iTRACK project partners, and its generated scenarios were shown to be diversified.

ACKNOWLEDGEMENT

This work is carried out as part of the iTRACK project, funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No 700510.

REFERENCES

[1] N. S. Noori, Y. Wang, T. Comes, P. Schwarz, and H. Lukosch, "Behind the Scenes of Scenario-Based Training: Understanding Scenario Design and Requirements in High-Risk and Uncertain Environments," in Proceedings of the 14th ISCRAM Conference, Albi, France, 2017, pp. 948–959.

[2] D. Mendonca, G. E. Beroggi, D. Van Gent, and W. A. Wallace, "Designing gaming simulations for the assessment of group decision support systems in emergency response," Safety Science, vol. 44, no. 6, pp. 523–535, 2006.

[3] T. Comes, N. Wijngaards, and B. Van de Walle, "Exploring the future: Runtime scenario selection for complex and time-bound decisions," Technological Forecasting and Social Change, vol. 97, pp. 29–46, 2015.

[4] V. A. Bañuls and M. Turoff, "Scenario construction via Delphi and cross-impact analysis," Technological Forecasting and Social Change, vol. 78, no. 9, pp. 1579–1602, 2011.

[5] T. Comes, N. Wijngaards, J. Maule, D. Allen, and F. Schultmann, "Scenario reliability assessment to support decision makers in situations of severe uncertainty," in Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), 2012 IEEE International Multi-Disciplinary Conference on, 2012, pp. 30–37.

[6] J. Z. Hernandez and J. M. Serrano, "Knowledge-based models for emergency management systems," Expert Systems with Applications, vol. 20, no. 2, pp. 173–186, 2001.

[7] "iTrack Project - iTRACK." [Online]. Available: http://www.itrack-project.eu/. [Accessed: 29-Jan-2018].

[8] T. Comes, N. Wijngaards, and F. Schultmann, "Efficient scenario updating in emergency management," in Proceedings of the 9th International Conference on Information Systems for Crisis Response and Management, 2012.

[9] F. A. O'Brien, "Scenario planning—-lessons for practice from teaching and learning," European Journal of Operational Research, vol. 152, no. 3, pp. 709–722, 2004.

[10] "IHP Training / Exercises." [Online]. Available: http://www.ihp.nu/training. [Accessed: 02-Feb-2018].

[11] G. Wright and P. Goodwin, "Decision making and planning under low levels of predictability: Enhancing the scenario method," International Journal of Forecasting, vol. 25, no. 4, pp. 813–825, Oct. 2009.

[12] P. J. Schoemaker, "Scenario planning: a tool for strategic thinking," Sloan management review, vol. 36, no. 2, p. 25, 1995.

[13] M. Comes, "Decision maps for distributed scenario-based multi-criteria decision support," Doctoral dissertation, Fakultät für Wirtschaftswissenschaften des Karlsruher Instituts für Technologie (KIT), Karlsruhe, Baden-Württemberg, Germany, 2011.

[14] S. Schnaars and P. L. Ziamou, "The essentials of scenario writing," Business Horizons, vol. 44, no. 4, pp. 25–25, 2001.

[15] I. S. Mayer, L. Carton, M. de Jong, M. Leijten, and E. Dammers, "Gaming the future of an urban network," Futures, vol. 36, no. 3, pp. 311–333, 2004.

[16] C. C. Hartog, "Scenario design for serious gaming," 2009.

[17] P. Thomas, J.-M. Labat, M. Muratet, and A. Yessad, "How to evaluate competencies in game-based learning systems automatically?," in International Conference on Intelligent Tutoring Systems, 2012, pp. 168–173.

[18] C. Pedersen, J. Togelius, and G. N. Yannakakis, "Modeling player experience for content creation," IEEE Transactions on Computational Intelligence and AI in Games, vol. 2, no. 1, pp. 54–67, 2010.

[19] "About the project | The Aid Worker Security Database." [Online]. Available: https://aidworkersecurity.org/. [Accessed: 03-Dec-2017].

[20] J. R. Quinlan, "Simplifying decision trees," International Journal of Man-Machine Studies, vol. 27, no. 3, pp. 221–234, Sep. 1987.

[21] "Global Terrorism Database." [Online]. Available: https://www.start.umd.edu/gtd/. [Accessed: 03-Dec-2017].

[22] "Training.dss.un.org - Online courses by the United Nations Department of Safety & Security," United Nations Department of Safety & Security. [Online]. Available: https://training.dss.un.org/. [Accessed: 06-Dec-2017].

[23] S. Bickley, "Security Risk Management: a basic guide for smaller NGOs," European Interagency Security Forum (EISF), 2017.

[24] W3Techs, "W3Techs - extensive and reliable web technology surveys." [Online]. Available: https://w3techs.com/. [Accessed: 26-Nov-2016].

[25] W3Techs, "Usage Statistics and Market Share of Server-side Programming Languages for Websites, November 2016," Nov-2016. [Online]. Available: https://w3techs.com/technologies/overview/programming_language/all. [Accessed: 26-Nov-2016].

[26] "R, Python Duel As Top Analytics, Data Science software – KDnuggets 2016 Software Poll Results." [Online]. Available: https://www.kdnuggets.com/2016/06/r-python-top-analytics-data-mining-data-science-software.html. [Accessed: 05-Dec-2017].

[27] "Graphviz - Graph Visualization Software." [Online]. Available: https://www.graphviz.org/. [Accessed: 02-Dec-2017].

[28] A. Zook, S. Lee-Urban, M. O. Riedl, H. K. Holden, R. A. Sottilare, and K. W. Brawner, "Automated Scenario Generation: Toward Tailored and Optimized Military Training in Virtual Environments," in Proceedings of the International Conference on the Foundations of Digital Games, New York, NY, USA, 2012, pp. 164–171.