Automated Scenario Generation
Coupling Planning Techniques with Smart Objects

Gwen R. Ferdinandus[1], Marieke Peeters[1,2], Karel van den Bosch[2] and John-Jules Ch. Meyer[1]
[1]*Information and Computing Sciences, Utrecht University, the Netherlands*
[2]*Training & Performance Innovations, TNO, Soesterberg, the Netherlands*
g.r.ferdinandus@gmail.com, {m.m.m.peeters, j.j.c.meyer}@uu.nl, karel.vandenbosch@tno.nl

Keywords: Automated Scenario Generation : Adaptive Educational Games : Serious Games : Scenario-Based Training

Abstract: Serious games allow for adaptive and personalised forms of training; the nature and timing of learning activities can be tailored to the trainee's needs and interests. Autonomous game-based training requires for the automatic selection of appropriate exercises for an individual trainee. This paper presents a framework for an automated scenario generation system. The underlying notion is that a learning experience is defined by the objects and agents that inhabit the training environment. Our system uses automated planning to assess the behaviour required to achieve the (personalised) training objective. It then generates a scenario by selecting semantically annotated (or 'smart') objects and by assigning goals to the virtual characters. The resulting situations trigger the trainee to execute the desired behaviour. To test the framework, a prototype has been developed to train the First Aid treatment of burns. Experienced instructors evaluated scenarios written by three types of authors: the prototype, first-aid experts, and laymen. The prototype produced scenarios that were at least as good as laymen scenarios. First-aid experts seemed the best scenario writers, although differences were not significant. It is concluded that combining automated planning, smart objects, and virtual agent behaviour, is a promising approach to automated scenario generation.

1 INTRODUCTION

Serious games have become increasingly popular as educational tools. Advances in graphic and AI techniques have provided us with virtual environments, inhabited by believable characters, where the trainee can practise the learning tasks autonomously. To enhance the effectiveness of these environments, serious games incorporate features from efficacious training forms (e.g. Peirce et al. (2008) and Peeters et al. (2012). A training methodology that, because of its story-like nature, lends itself especially well for this prupose is Scenario-based Training (SBT). SBT primarily concentrates on the type of exercises, i.e. contextualized, whole-task storylines, exemplifying the learning-by-doing approach to training (Oser et al., 1999; Salas et al., 2006). In SBT the trainee is confronted with a representative sequence of events (the scenario) within a simulated environment (e.g. the game). Of course, most training methodologies also recommend a certain ordering of learning tasks (Merrill, 2002). However, the planning and ordering of learning tasks is not the issue in the work presented here.

The problem discussed in this paper is that once such an ordering has been established, suitable exercises need to be created to provide the trainee with practice opportunities regarding the selected learning task. Since manual scenario creation is a time-consuming process, most systems reuse a limited set of scenarios linked to each learning task, however, after several occasions, scenario repetitions becomes inevitable. This is a problem for training directed at skill maintenance, i.e. continued training. Moreover, to ensure effectiveness, the exercises (scenarios) should be adapted to the individual needs and abilities of the trainee, and offer him varied experiences (Peeters et al., 2012). As such, the need for automated scenario generation arises.

In this paper we propose an automated scenario generation framework to produce training scenarios that encompass a previously selected learning objective while warranting complete and coherent storylines. The next section first considers related work on automated scenario generation. Section 3 then details the design, followed by the evaluation in Section 4. Finally, Section 5 discusses the implications of our research and opportunities for future work.

2 RELATED WORK

There is no generally accepted solution to scenario generation, despite it being a growing subject of interest. This section discarusses some existing approaches.

The two works discussed below focus on the creation of a game world that matches the intended learning task. Martin et al. (2009) propose to construct an initial scenario based on the training task, which in turn is extended by adding events to increase the complexity level. The requirements on the game world that follow from the resulting conceptual scenario are addressed using a shape grammar. Lopes and Bidarra (2011a) also focus on the realisation of the scenario within the virtual world, arguing that the contents of the game world determine the trainee's experience. They propose the use of Smart Objects (Kallmann and Thalmann, 1998), which are annotated with the services they offer to their surroundings, such as the experiences they could offer a player. These annotations can be used to steer the content selection process. Although in both of these approaches content generation sprouts from the initial learning objective, the lack of an explicit task representation makes it difficult to interpret the resulting training scenario within the context of the training domain. It is also impossible to derive the expected action plan for the trainee.

In contrast, Niehaus and Riedl (2009) employ automated planning techniques to construct a scenario based on the trainee's expected action sequence. By adapting a default scenario plan, consisting of ordered high level tasks, these action sequences can be adjusted to the needs and abilities of the trainee while maintaining a coherent storyline . An important advantage of this approach is the possibility to track the actions the trainee is required to perform to accomplish the learning task. Moreover, the action sequence has been derived from an explicit representation of the training domain, which promotes the interpretation of the scenario with respect to the learning content.

The next two approaches select events to persuade the trainee into performing the desired actions. Grois et al. (1998) employ probabilistic networks to compute a set of events likely to cause an opportunity for practising the learning task. Zook et al. (2012) use a basic set of events and, subsequently, use a genetic algorithm to extend, mutate and improve the sequence of events until an acceptable scenario has been generated. Both approaches offer interesting alternatives (or additions) yet require very specific data such as probability functions and quantitative scenario evaluation functions, that are all but trivial to define. None of the last three approaches discuss how the scenario should be facilitated by the game world.

3 DESIGN

This section describes the Scenario Generator framework. First, we consider the requirements for effective training scenarios that have driven the design process. Then, the general design is presented, followed by an explanation of the framework's two main components: the Action Planner, and the Object Selector.

3.1 Scenario Requirements

To understand which elements determine the effectiveness of a training scenario we have gathered information from literature studies (Martin et al., 2009; Issenberg et al., 2005; Peeters et al., 2012) and interviews with experienced instructors from different training domains (i.e. First Aid, In-Company Emergency Assistance, and the Dutch Royal Navy). From this research, we concluded that in order to be effective a training scenario must be 1) focused on the learning objective, 2) adapted to the trainee's competency level, 3) representative of real life situations, and 4) complete with respect to high level procedures. The last requirement is illustrated by the following example: a first-aid training exercise should not be restricted to the treatment of the victim, but each exercise should also contain the preceding steps of securing the environment and determining the problem, since real life situations will never encompass merely the treatment of the victim.

3.2 The Scenario Generator

The Scenario Generator has been designed to work within the context of an agent-based Adaptive Educational Game (AEG). Figure 1 depicts the general design of the Scenario Generator within the context of the AEG. The system consists of two main parts: an Action Planner and an Object Selector. The Action Planner uses the learning task provided by the AEG to generate a complete and coherent action plan for the trainee (see Section 3.2.1). The resulting action plan imposes requirements on the game world; e.g. actions may require the presence of objects or coordination with tasks performed by other characters (agents). The Object Selector warrants that such requirements are met by selecting the appropriate objects and agents (see Section 3.2.2).

The Scenario Generator requires the following input from the AEG. 1) The *learning task* encompassing the learning goal that the trainee is supposed to practise during the scenario (e.g. 'treat burn'). 2) The *difficulty level* at which the learning task should be

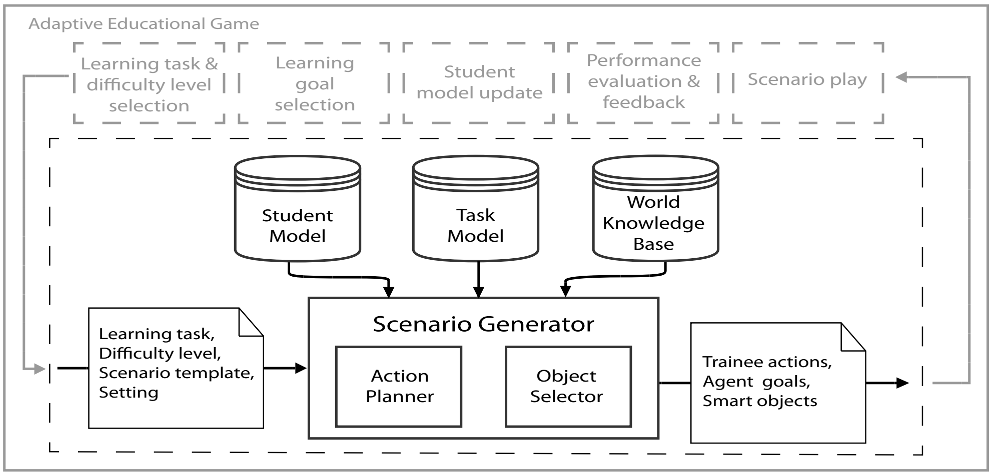

Figure 1: An overview of the Scenario Generator within the Adaptive Educational Game.

practiced. This level is defined as a value between 0 and 1, representing the skill level required to successfully perform the learning task, where 0 indicates no skill and 1 represents mastery. 3) The *scenario template*: an ordered list of high-level tasks that constitute a complete training exercise. 4) The *setting*, identifying the desired contextual location of the scenario (e.g. 'the kitchen'). The setting is used to ensure authenticity and influences the object selection process. Additionally, the Scenario Generator is assumed to have access to a student model containing the trainee's current performance levels, a domain-specific task model containing information about the decomposition of learning tasks, and a world knowledge base describing the available objects and agents along with their domain-specific features.

The output of the Scenario Generator is a *scenario plan* that contains all the information required by the AEG. 1) *Trainee Action Plan*: a partially ordered list of actions the trainee is expected to perform. 2) *Agent Goals*: a list of high-level goals for the virtual agents (e.g. 'create fire'). 3) *Smart Objects*: a list of the required Smart Objects annotated with parameters that influence the difficulty level at which they offer the required services (i.e. interaction possibilities).

3.2.1 Action Planner

The Action Planner creates a coherent action sequence for the trainee that encompasses the learning task, and constitutes a complete training exercise because of the scenario template used in the process. In addition, the planner determines the goals for the virtual agents (i.e. events) that are expected to trigger this action sequence. Like Niehaus and Riedl (2009), the action planner employs a hybrid HTN plan-space planner. Such a planner employs domain-specific knowledge to decompose abstract high-level tasks into concrete actions meanwhile addressing open preconditions by adding new actions and introducing causal links between existing actions. The domain-specific knowledge used by the HTN planner is stored in the form of so-called *methods*. A method specifies how a high-level task can be decomposed into subtasks (see Ghallab et al. (2004) for a more formal definition). For our particular purpose, we extend the domain knowledge stored in an HTN method with two components: preconditions in the form of services that must be offered by the game world before a method can be applied; and an indicator of the difficulty level of the method. An example of a precondition would be that to decompose the task 'remove danger' into 'extinguish fire' the service 'fire' must first be offered. The difficulty level, a value between 0 and 1, allows the Action Planner to influence the difficulty of the scenario by comparing applicable methods in terms of complexity. For example, decomposing the task 'treat arm injury' into 'clean wound' and 'dress wound' might be less complex than decomposing the same task into 'clean wound', 'dress wound' and 'apply splint'. The difficulty level is determined by domain experts. The following paragraphs will continue the description of the action planner by detailing the different steps of the scenario generation process (also depicted in Figure 2).

Incorporate the learning task. To ensure a complete training exercise, the scenarios need to contain the learning task and follow the scenario template. Therefore, the planner searches for (sequences of) HTN methods that can be applied to decompose the high-level task(s) from the scenario template into a series of subtasks containing the learning task.

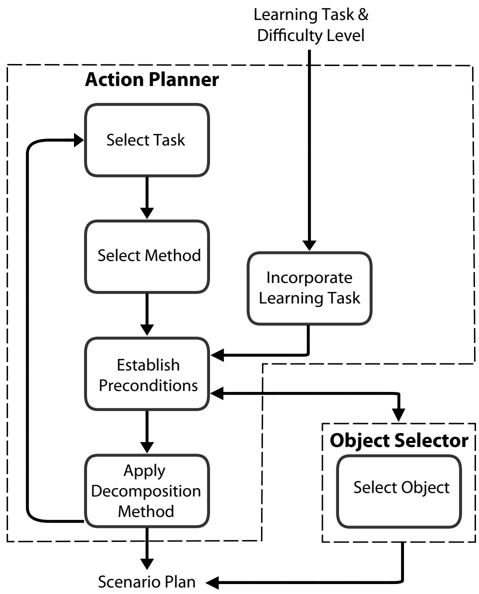

Figure 2: Flowchart depicting the planning process of the Scenario Generator.

Select task. Once the learning task has been introduced in the planning, the system iteratively addresses the remaining high-level tasks in the scenario template. During this process the system randomly selects the next task to work on in order to promote variety between scenarios with the same input.

Select method. Each selected task is decomposed into actions by applying decomposition methods. This process is guided by two considerations: the preconditions already established by the game world so far, and the difficulty level of the method. First, the planner selects the methods with the highest number of established preconditions thereby ensuring the coherence of the scenario. If there are multiple candidates, the selection process is guided by the smallest difference between the method's difficulty level and the desired difficulty level. If the task at hand is the learning task, the desired difficulty level is received from the AEG, otherwise the desired difficulty level is retrieved from the student model. Any remaining ties between methods are broken randomly.

Establish preconditions. Before a method can be applied, all its preconditions (required services) must be fulfilled. The Action Planner forwards the service requests to the Object Selector. Sometimes a service can only be offered after a certain goal has been achieved, e.g. a 'fire' can only be offered after it has been ignited. The Object Selector returns these goals to the Action Planner which plans an action sequence for a virtual agent to achieve this goal. NB: Although the action sequence is not relevant for the output (the virtual agents require goals only), it must be computed since the actions may in turn require new services from the game world.

Apply decomposition method. The final step in the process is to actually replace the high-level task with the subtasks specified in the selected method. If these subtasks are actions, the Action Planner also needs to ensure any open preconditions of the actions.

3.2.2 Object Selector

The Object Selector is grants the service requests posed by the Action Planner, by reasoning about the services (i.e. interaction possibilities) the available Smart Objects can offer, such as a match offering the service 'fire'. Agents are considered to be a special kind of Smart Object that can offer more complicated services. In addition, Smart Objects are annotated with preconditions for their offered services, an indication of their belonging in specific settings, and a difficulty level. The preconditions can be other required services, or goals that should be achieved by an agent. The difficulty level is based on three aspects: the complexity of its use, the obviousness of its intended use for the service, and the adaptability of its difficulty level (i.e. a fire can be small and controllable or a raging inferno). The Object Selector determines the most appropriate object based on its belonging in the provided scenario setting and its match to the desired difficulty level. If the selected object requires any other services, the Object Selector iteratively fulfils these requests; if the object requires the achievement of specific goals the Object Selector forwards the request to the Action Planner.

4 EVALUATION

A (proof of concept) implementation of the scenario generator was evaluated, be it with some additional simplifications due to time and resource constraints. The prototype employs a rudimentary HTN planner (not an HTN plan-space planner); all service preconditions are defined as actions (not goals), and there are no ordering constraints on the actions of the agents. These simplifications are of no great concern for the current test setup, though should be addressed in future implementations to fully exploit the advantages of using autonomous agents. The prototype is developed for the training domain of First Aid, which has the advantage that it requires no complex world representations and has clearly defined procedures. To further limit the size of the knowledge base the prototype was restricted to burn-related incidents.

Table 1: Data exploration: mean scores (and standard deviations) over all raters

Dependent Variable	Scenario Source			
	Expert	Layman	System	Overall
competency suitability	.867 (.448)	-.061 (.431)	.000 (.527)	.269 (.413)
task suitability	1.933 (.290)	.389 (.453)	.800 (.534)	1.041 (.358)
authenticity	1.733 (.210)	.450 (.400)	.583 (.199)	.922 (.167)

4.1 Method

Evaluators - oblivious of the research question - rated scenarios written by the system, human experts and laymen in random order. The scenarios were evaluated on three of the requirements identified in Section 3.1: suitability for the learning task (*task suitability*), suitability for the trainee's competency level (*competency suitability*), and *authenticity*. Completeness of the exercise was omitted, it being too hard to recognize by just the scenario description (the trainee's expected action plan should be included). It was hypothesised that for all dependent variables the experts would score best, followed by the system followed by the laymen. The setup was a within-subjects design; all evaluators rated all scenarios from all sources.

Participants. The experiment used 5 evaluators (all First Aid instructors), and 9 writers (5 First Aid instructors and 4 laymen).

Scenarios. The test set consisted of 36 scenarios: 12 scenarios for each source (experts, laymen, and system). All authors wrote scenarios based on 3 features: 1) the learning task (i.e. treat burn, calm victim, or ensure ABC), 2) the trainee's competency level (i.e. beginner or advanced), and 3) the setting (i.e. home, restaurant, laboratory, or park). Counterbalancing ruled out any possible effects resulting from these features. The authors used a predefined format, consisting of the background story (i.e. what happened), instructions for the 'victim agent', and a list of required objects. The output of the system was not in natural language and was manually rewritten following predefined translation rules.

Questionnaires. Each page of the questionnaire contained a scenario description followed by 3 7-point Likert-scale questions. The *task suitability* and *competency suitability* were measured indirectly; the evaluators rated the suitability of the scenario for a beginner and for an advanced trainee. The highest of the two was used to represent the *task suitability*. The score of the question matching the intended difficulty level was used as a measure of the *competency suitability*.

Procedure. In advance of the experiment proper and after an extensive instruction, inter-rater reliability was fostered by a joint discussion on 2 sets of 6 example scenarios.

Results. Table 1 shows the means and standard deviations of the test set. The results follow the hypothesised trend with the experts scoring highest followed by the system followed by the laymen. The table also shows large standard deviations, in particular for the *system* and *layman* scenarios. The intraclass correlation coefficient (ICC) using the 2-way random model suggested substantial agreement (r=0.732; p<.001). Any missing values (0.01% of the 540 values) were imputed using the SPSS expectation-maximisation procedures (Little and Rubin, 1989). A repeated measures ANOVA revealed significant differences between the sources for *task suitability* ($F(2,8) = 6.699; p = .020$) and *authenticity* ($F(2,8) = 6.220; p = .023$), but not for *competency suitability* ($F(2,8) = 3.529; p = .080$). Post-hoc tests using the Bonferroni correction procedure revealed no significant differences between the sources in one on one comparisons.

Discussion. The experiment revealed no significant differences between the scenarios coming from different sources (i.e. experts, laymen, and system). One possible explanation might have been the large standard deviations, which could have been caused by disagreement among the evaluators, however, the ICC analysis rules out that possibility. Several points for discussion that may shed some light on these results are discussed. First of all, the settings were purposefully varied to show that the prototype is capable of generating scenarios for various situations. However, these settings may have forced the experts to write scenarios for settings they normally would not use. Secondly, the current template does not include the unfolding of the scenario, nor the trainee's expected action sequence, thereby failing to reveal the source's intentions underlying the produced scenarios. The evaluators may have used their experience to interpret the scenario descriptions to infer these aspects, covering up any possible differences. A final point of attention is the importance of involving domain experts in the development of the knowledge base: the task decompositions and the objects used. Even though the used knowledge base contained accurate information, the instructors indicated that it contained some content they would never use in their own scenarios, e.g. electrical burns.

5 DISCUSSION & CONCLUSION

This paper addressed the issue of automated scenario generation within the context of an Adaptive Educational Game (AEG). The proposed framework integrates a hybrid HTN planner to plan the trainee's actions with a content selection mechanism based on Smart Objects to control the realisation of the scenario within the game world. This results in a separation of the action plan construction - enabling the retrieval of the scenario's underlying didactic intentions - and the game world creation - enabling the use of a separate smart objects database which is easily extended, in contrast to hard-coded objects in the planner itself. Although the evaluation experiment did not provide significant results, the authors are hopeful that further experimentation will provide more definitive answers.

Several directions for further research can be suggested. First of all, the experiment showed that some of the desired functionalities of the system interfered with its primary goal of producing scenarios with a complexity level that fits the skill level of the trainee. Since the prototype generated complete training exercises, additional tasks were addressed in the scenario on top of the learning task. Because a global difficulty level was used, the trainee is expected to perform the additional tasks at the same difficulty level as the learning task. This combination of multiple tasks seemed too much to handle for the trainee. Two possible (and complimentary) solutions are the integration of a more fine-grained difficulty control system such as for example a performance curve suggested by Zook et al. (2012) and the introduction of 'colleague agents' that take over part of the responsibilities of the trainee. A second suggestion for further research is the comparison of the different approaches proposed for automated scenario generation. So far it has been difficult to compare different approaches since each system uses its own standards and criteria. However, a comparison might be highly informative and show the strengths and weaknesses of the different approaches, possibly leading to hybrid solutions.

To conclude, the framework presented here is a first step towards a system that can generate effective and personalized scenarios automatically. In the future, attaching the system to a game-based training program will offer trainees extensive access to high-quality training opportunities.

REFERENCES

Ghallab, M., Nau, D., and Traverso, P. (2004). *Automated Planning: Theory & Practice*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Grois, E., Hsu, W. H., Voloshin, M., and Wilkins, D. C. (1998). Bayesian network models for generation of crisis management training scenarios. In *Proc. IAAI-98*, volume 10, pages 1113–1120. AAAI.

Issenberg, S., McGaghie, W., Petrusa, E., Gordon, D., and Scalese, R. (2005). Features and uses of high-fidelity medical simulations that lead to effective learning: a beme systematic review. *Medical Teacher*, 27(1):10–28.

Kallmann, M. and Thalmann, D. (1998). Modeling objects for interaction tasks. In *Proc. Eurographics Workshop on Animation and Simulation*, volume 9, pages 73–86. Citeseer.

Little, R. and Rubin, D. (1989). The analysis of social science data with missing values. *Sociological Methods & Research*, 18(2-3):292–326.

Lopes, R. and Bidarra, R. (2011). Adaptivity challenges in games and simulations: A survey. *IEEE Transactions on Computational Intelligence and AI in Games*, 3(2):85–99.

Martin, G., Schatz, S., Bowers, C., Hughes, C. E., Fowlkes, J., and Nicholson, D. (2009). Automatic scenario generation through procedural modeling for scenario-based training. In *Proc. HFES-09*, volume 53, pages 1949–1953. SAGE Publications.

Merrill, M. D. (2002). First principles of instruction. *ETR&D*, pages 43–59.

Niehaus, J. and Riedl, M. O. (2009). Scenario adaptation: An approach to customizing computer-based training games and simulations. In *Proc. Workshop on Intelligent Educational Games (AIED-09)*, volume 3, pages 89–98.

Oser, R. L., Cannon-Bowers, J. A., Salas, E., and Dwyer, D. J. (1999). *Enhancing Human Performance in Technology-Rich Environments: Guidelines for Scenario-Based Training*, volume 9 of *Human/Technology Interaction in Complex Systems*, pages 175–202. Jai Press Inc.

Peeters, M., Bosch, K., Meyer, J. J., and Neerincx, M. (2012). Situated cognitive engineering: the requirements and design of directed scenario-based training. In *Proc. ACHI-12*, volume 5, pages 266–272. XPS.

Peirce, N., Conlan, O., and Wade, V. (2008). Adaptive educational games: Providing non-invasive personalised learning experiences. In *Proc. DIGITEL-08*, volume 1, pages 28–35. IEEE.

Salas, E., Priest, H. A., Wilson, K. A., and Shawn Burke, C. (2006). *Scenario-Based Training: Improving Military Mission Performance and Adaptability*, volume 2 of *Military Life: the psychology of serving in peace and combat*, chapter 3, pages 32–53. Praeger Security International.

Zook, A., Urban, S. L., Riedl, M. O., Holden, H. K., Sottilare, R. A., and Brawner, K. W. (2012). Automated scenario generation: toward tailored and optimized military training in virtual environments. In *Proc. FDG-12*, volume 7, pages 164–171. ACM.