An Authoring Tool for modeling the Narrative Flow in a Virtual training Application

Nahum Alvarez and Helmut Prendinger

National Institute of informatics, Tokyo, Japan
{nahum,helmut}@nii.ac.jp

Abstract. The use virtual environments in training applications has the potential to provide rich training experiences to students, but it is difficult to design and manage such training sessions in real time due to the number of parameters to pay attention to: timing of events, difficulty, user's actions and their consequences or eventualities are some examples. Narrative driven scenario managers confront this issue by generating dynamically these variables, making different each time the user plays a scenario. However, leaving the control to such systems makes the training difficult to predict, and in the case of training or learning applications the designer may have specific goals to introduce in the session. For that purpose, we have designed an authoring tool for our virtual narrative application used for training biohazard procedures. Our system contains a Narrative Manager that controls the simulation deciding which events will take place, and when, by controlling the narrative balance of the session. Our authoring tool allows the designer to specify a narrative curve and the Narrative Manager adapts the narrative flow of events to that curve. Thus, we provide a method to model the training's narrative intensity without having to specify the ordering of each concrete event or numerical parameters each time we want to control some aspect of the dynamic generated session.

Keywords: Interactive Storytelling, Authoring Tools, Virtual Training Applications, Drama Modelling.

1 Introduction

Recently, narrative techniques are becoming more common in a wide range of applications, especially in virtual training simulation or e-learning applications: we can see examples in the works of Rizzo et al. [1] where they present a clinical diagnosis training system including virtual agents with a scripted backstory and speech recognition for simulating patients, the one in Reger et al. [2] using a virtual reality simulation to treat posttraumatic stress disorder in soldiers or in Raybourn et al. [3], describing a multiplayer simulation system for training army team leaders. In order to allow a rapid design and optimal management of the narrative scenarios, authors have answered this issue with different techniques: Carpenter et al. [4] proposed an approach that uses branching trees to advance in a story in which the user has to take decision to manage a crisis. Another work [5] describes a protocol to create a branching structure in which training scenarios are constructed like storyboards, and applies it for a web-based application. The approach of Ponder et al. [6] uses a human as a scenario director controlling events in the virtual environment. All of these systems present a well-known problem of the branching approach: it does not scale well when the number of events or possibilities in the story increases and there is no perfect solution for rapid authoring of drama games [7]. In order to solve this issue is not unusual to incorporate a drama manager to control the events in the virtual world. For example, the commercial software "Giat Virtual Training®" includes a scenario engine driven by scripts that control the virtual world happenings [8]. Other drama manager can be found in the application presented by Habonneau et al. [9], an interactive drama application for teaching how to take care of brain-damaged people. This method proved to be very immersive and engaging for the users, allowing them to find themselves in a realistic environment where unexpected things can happen, but still it wasn't evaluated in terms of learning, and it does not tutors the user at all, being an experience only narrative without training elements in which the user interacts with characters and see what happens.

In this paper we focus in authoring tools for training scenarios, with special attention in narrative driven ones. We present an authoring tool for modulate the drama in virtual scenarios simulated by our system, a Narrative Manager tested with good results in a virtual training application for teaching bio-safety procedures, called Bio-Safety Lab. As we explained, including a drama manager is a good solution for the problem of authoring virtual learning or training sessions; however it adds another necessity of authoring. Instead of the scenario, in this case it is the tutor system the one that we would like to design or customize in an easy way, especially in the case of learning or training applications because the scenario designer may have some requirements for the training session: for example we can require stressing the user on some specific point of the session and next leaving him to relax with easy to solve events. By implementing a good authoring tool, the training expert can control directly the tutor system and the scenario without having to know scripting or programming languages, also, the system will adapt better to its users [10]. The rest of this document is structured in four chapters: Section 2 reviews the current state in authoring tools technologies for training and learning applications. In Section 3 we describe briefly our training application, Bio-Safety Lab, in order to put the reader in context for Section 4, where we present our authoring tool for modulate the narrative sessions in our training system. Finally, in section 5 we discuss the advantages and limitations of our approach, and present our conclusions and next steps in the topic.

2 Related Work

Commonly, in simpler learning applications that work as a series of self-assessment tests, a narrative scenario director is not necessary and the authoring tool consists in a simple graphic editor with drag and drop features. Interestingly, an Intelligent tutor authoring tool that uses crowdsourcing for creating its knowledge database showed that even if the results are not bad with this strategy and is faster to create a database this way than using a domain expert, has a low recall and is less reliable [11]. It seems that the help of a domain expert is necessary for creating good knowledge bases, so it is important to have a complete but easy to use authoring tool in order to minimize the deploy of the training or learning system. For example, Tripoliti et al. [12] presents a complete tool for creating tutors for medical diagnosis training. The tutor works as a series of steps that have to be completed, and the tool allows customizing the steps in a graphic editor with drag and dropping features. Another work [13] presents a tool for designing scenarios for a physics game with real time interaction in 2D scenarios, so is more complex than a set of self-assessment tests. The scenario editor goes further than the previous example in the usability aspect, allowing playing the scenario inside the editor in order to fine tuning its features. However, the increased complexity of learning applications requires a more careful scenario design: impossible or unplayable scenarios can be created or simply they can be unsatisfactory or boring (too easy, too difficult), so the designer has to test all the details and possible situations. This problem can be solved with the addition of a scenario director layer controlling the training, like our Narrative Manager, allowing the scenario designer to not to worry about if the training is too easy or not. A more complex tutor is described in the Chocolato system [14], using ontologies and allowing the selection of the type of pedagogic theory used and the user's goals. In this case, the authors confirmed than the authoring tool effectively improves the performance, saving efforts and time to the domain experts and allowing a rapid deployment of the learning system. Other example can be found in the work of Karampiperis et al. [15] describing the ER designer toolkit which essentially is a trigger listener editor. Its graphic environment generates the code necessary to define the virtual tutor behavior. However, they don't complete the tool with a responsive training environment, so the human domain expert has to decide what to do when the virtual tutor detects the user's actions thus lacking a layer for controlling the simulation (like our Narrative Manager), with an authoring tool to define how this control layer will act. We can find another advance tool with more common points with our system [16] which presents an authoring tool for an e-learning platform. Though the system presents a simple form of training (only present the user problems with only one answer en a series of self-assessment exercises), the customization capability is wide, allowing the selection of the resources used for the learning and how they will be used. This platform have several similarities to our Bio-Safety Lab, in the sense that they have different layers of knowledge, separating the raw data containing information about exercises, defined in a knowledge base by an expert, from the resources that define an scenario, describing what data is used and how they will be used in the learning. Also, the authoring tool allows a high grade of variability because the scenario designer does not specify all the details of the session, but only give a pattern of how the tutor will act. This is again a common point to us, because our Narrative Manager acts as an independent entity, only requiring general directions from the scenario designer to act. However, the system is not fully operational and it was not tested. Other interesting tool is seen in Olsen et al. [17] including a drag & drop interface and a knowledge model based in "behavior" graphs, allowing editing the nodes and links graphically, similar to the structure we use. It allows branching and collaborative work between users, but our model is more sophisticated, containing different types of parameters and groupings. Other example of a training system with high level author capabilities can be seen in Sottilare et al. [18], but again, it is not fully operational yet, showing examples like a thermodynamics tutor for undergraduates [19]. The application allows designing virtual tutors by creating behavior rules about user actions, customized questionnaires and profiling the users. The system can be plug to other simulation frameworks [20], in this case Unity 3D, like our Bio-Safety Lab. Creating behavior rules is a good way to design the tutor behavior. We can see this strategy taken to a step further in the CTAT system [21]. This generic virtual tutor designer uses jess rules and a database with "memory elements" for defining the goals of the users, and even the graphic interface of the learning application. Similar to the ASTUS system, allows designing almost any kind of tutor; in a sense both are equivalent, as seen in [22]. However, its authoring language is not accessible for non- programmers, going as far as requiring an authoring tool for the authoring tool. In our case, the authoring tool contained in Bio-Safety Lab is easy to access and allows the designer to model the tutor language. Finally, when the training applications have special requirements i.e. a collaborative tool, the authoring system needs some additional features. An example can be again the CTAT system, which needs some adaptation to suit the requirements for collaborative learning applications [23], or in the case of augmented reality learning applications, which add the tagging and optic recognition requirements of this kind of systems [24].

3 Narrative manager

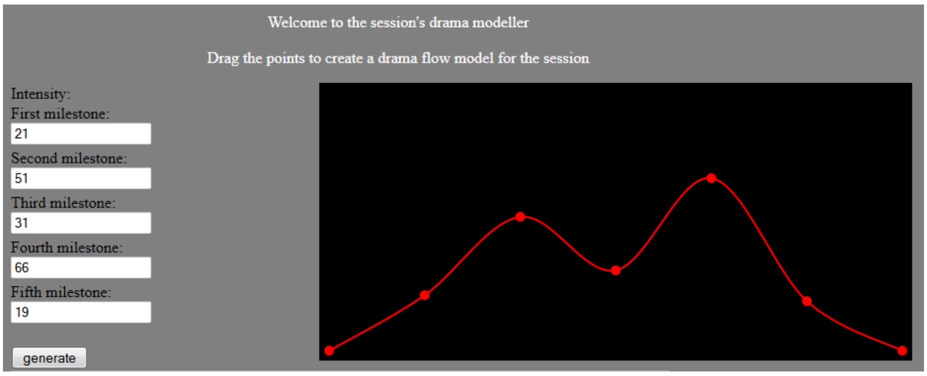
In our training system, the user takes the role of a bio-safety laboratory worker who has to do some experiments. When an accident happens, involving the spill of a contaminated human blood sample, the user has to follow the official protocol to clear it. The application implements a Narrative Manager, a control module that that manages the scenario's events using a narrative model which optimally selects the timing and the event to trigger. We describe it in detail in Alvarez et al., 2014 [25] (under re-submission) but for context purposes we will summarize its features in this section. Our Narrative Manager decides how and when certain events will happen in the scenario, to either decrease task difficulty if the user has trouble completing it, or to increase task difficulty if the task is too easy for the user. The ultimate goal of the Narrative Manager is to engage the user in an interesting narrative entangled within the training process, maximizing the number of events for the user to solve, and thus increasing the possibility of learning acquisition. The Narrative Manager uses a set of parameters that describe the dramatic value of the events and the situation. These parameters are used to decide which event will be triggered next and are inspired on Ware and Young's dimensions of conflict [26]. We use four parameters with our own nomenclature:

- Balance: an integer number that describes how beneficial or prejudicial an event is for the user (positive if it makes the task easier for the user and negative if it is difficult or hinders his performance). Each event contains its own balance value stored in the Task Tree, and the value is given by the domain expert who designs the tree. For example, breaking a bottle that contains an infectious sample would have a Balance of -6, and giving the user a hint has a Balance of +4.
- Global Balance: The Narrative Manager also maintains a global value for the accumulated balance of the whole training session by adding the balance of the triggered events. This Global Balance has a slow attenuating effect with time, tending towards 0. The rationale behind this attenuating effect is that receiving some hints or difficulties can affect the user in the short term but as time goes by this effect disappears: the hints are useful only for the immediate task and the problems can be resolved for the user if he/she takes some time to think.
- Impact: a positive value that represents the degree of dramatic load that an event has for the user when triggered. An event with a Balance value very near 0 can have high Impact if it has a strong visual or psychological effect. For example, when a toxic spill overflows its borders, it has a Balance of -2, being not very bad for the user, but is very startling for him, having an Impact of 5.
- Intensity: a global value maintained by the system that describes how dramatic the current situation is for the user. At the beginning its value is zero and it is updated dynamically by the Narrative Manager by adding or subtracting the Impact value of the triggered events to its previous value. The value is added or subtracted depending on its Balance value sign.

As we can see, the values of the Balance, Impact are static and quite subjective, since they are related to the dramatic quality of only one event. The domain expert has to define them by judging each event when he designs the scenario. The Narrative Manager will use these values for calculating the value of the related dynamic parameters, Global Balance and Intensity, that not only depend of the static value of the two static parameters but also the timing and the kind of the user events, having in account the previous actions of the user. Whenever the Narrative Manager is called it selects an event from the Task Trees which maximizes Impact, with the current Intensity as the upper boundary, and with the Balance most similar to the current Global Balance, but with opposite value. We use Intensity as an upper boundary because it represents the maximum dramatic load we allow for the user at one time, thus limiting the events that can happen. Also, the system has to move the Global Balance towards zero, so the desired event should have a Balance similar to the current Global Balance, but with opposite value. Using both parameters allows maintaining a balanced difficulty while at the same time creating events that impact the user in a dramatic way. The Narrative Manager was tested in an experiment with 30 users and obtained good results improving user's interest as well as their knowledge acquisition. Also, by using this module we make unnecessary to script the whole scenario each time we want different events, because it is the Narrative Manager the one in charge of deciding which events will trigger, making each session different. The scenario designers or the domain expert only have to define in an xml the events they want to input in the system and the values of the narrative parameters, so the system escalates reasonably well.

4 Authoring the Drama Manager Behavior

By having the Narrative Manager controlling the details of the training session (in our case, the events that happen in the virtual environment), we already have made easier the task of authoring the scenario: the knowledge expert simply has to define the possible events in a scenario and specify their narrative parameters of Impact and Balance. However, this feature has a downside: it is difficult to control or predict how the session will develop. It seems to be counterintuitive to want to control the events when we just have stated that not having to do so is an advantage, but is desirable to give to the domain experts some degree of control over the details of the training, especially if they want to reinforce certain behavior in the users or to teach them some concrete practice. Making simulations of the system can give a general idea of how the session will be, and giving careful values to the events' parameters can increment the probability that some concrete events will happen. However, the session always will try to maximize user engagement by keeping him challenged continuously. This can be sometimes contrary to the desires of the session designer: maybe is better for the user to put him in a more forgiving session because he previously finished a more difficult one and he has to reinforce some procedures. Other example would be when the experiment designer wants to limit some difficult events for some concrete moments: by giving a high impact value the designer only will know that the system will trigger that event later, but that is the only control he will have.



Welcome to the session's drama modeller

Drag the points to create a drama flow model for the session

Intensity:
First milestone: 21
Second milestone: 51
Third milestone: 31
Fourth milestone: 66
Fifth milestone: 19

Fig. 1. The Intensity Curve modeller tool for Bio-safety Lab. Moving the red dots in the curve the scenario designer can model how the Narrative Manager will behave in the sessions.

In order to answer this need, we have implemented an additional feature in Bio-Safety Lab: the Intensity curve authoring tool. This application allows drawing a curve that will be used as a reference for the Narrative manager when it selects the events in the session. A screen capture of the authoring tool can be seen in Figure 1. The graphic interface consists of a window in which the scenario designer can model a curve that represents the Intensity over the time in the training session. The curve contains five dots we called Milestones that can be dragged up and down with the mouse. While moving a milestone, its numerical value will appear in the textbox related to that milestone. This value represents a percentage of how near can be the intensity in that relative moment from the maximum possible Impact from the scenario's events. As we see, there is no time specified in the graph. The reason is because each session will likely have a different duration, and the only significant variable that we can detect is the events themselves.

Using as an example a session controlled by the curve depicted in Figure 1, when the system selects an event with an Impact near to the half of the maximum Impact event available, the system will suppose that has passed the second milestone (with a value of 51%), then the system will try to decrease the intensity to a third of that maximum value, reaching the third milestone (value of 31%) when doing that. In the example we decided using only five milestones, but we can add more dynamically. However, we think five milestone is enough, because this implies that we model the tendency of the Intensity of each fifth of the session, and we consider that an enough degree of control for the scenario designer. By adding more milestones the designer would have too much control in the scenario, effectively scripting the whole session and not allowing the Narrative Manager to control freely the Intensity curve.

The Narrative Manager can work with or without a given Intensity curve. If the scenario designer does not create one, the Narrative Manager will proceed as we described in the previous section, but when using one as a model, there are two changes in the process of event management: when the Intensity value is increased over the time and when a scenario event is selected by the Narrative Manager. In the first one, the Narrative Manager will use the next milestone value as a modifier for increasing the Intensity gradually. For example, if the milestone has a value of 100 (the maximum possible), the Intensity will increase with the time a 110% more than the usual value, and with a value of 0, the Intensity at only a 10% of its rate, and finally if the next milestone has a value lesser than the current Intensity, it won't increase with the time until its value reach the milestone. The rationale behind this variation in the rate of increasing the intensity is done because we want to increase it faster if the slope of the curve is very pronounced. There are other methods for varying the increment over time of the Intensity, but we decided use this one because its simplicity: a milestone in the top of the graph doubles the rate, and one in the bottom almost negates it, while a milestone in the middle, does not modify it. Then, the second change in the process happens when selecting the next event to trigger: as we described, a milestone's value is mapped to the maximum Impact of the available events in the scenario and the system uses it as a threshold. If the next milestone has a value bigger than the current Intensity, the Intensity will increase until an event with this value is triggered. Until that instant, other events can be triggered by the system, but if those events would reduce the Intensity value (due to having a Balance with negative sign), the Narrative Manager won't decrease it. When the triggered event has an Impact greater than the milestone value, the system will start using the next milestone's value (if there is no next milestone, it simply doesn't change). In the case of a milestone that is lesser than the current Intensity, then it means that the curve as a descending slope, so the system will not increase the intensity over time until an event with Impact greater than the milestone's value is triggered. The events triggered until then can only decrease the Intensity, and in case it is an event that should increase it, the system won't do it. With this two differences in the event management, the Narrative Manager can use the Narrative curve to create a similar Intensity curve for the session. The session will behave using the guidelines of the predefined curve, but it won't be exactly the same. This effect is intended: it is not our intention to give detailed orders to the Narrative Manager of how has to act, but only give it a general direction of how we want the training session. This way, each session will be still different even if we use the same Intensity curve to define it.

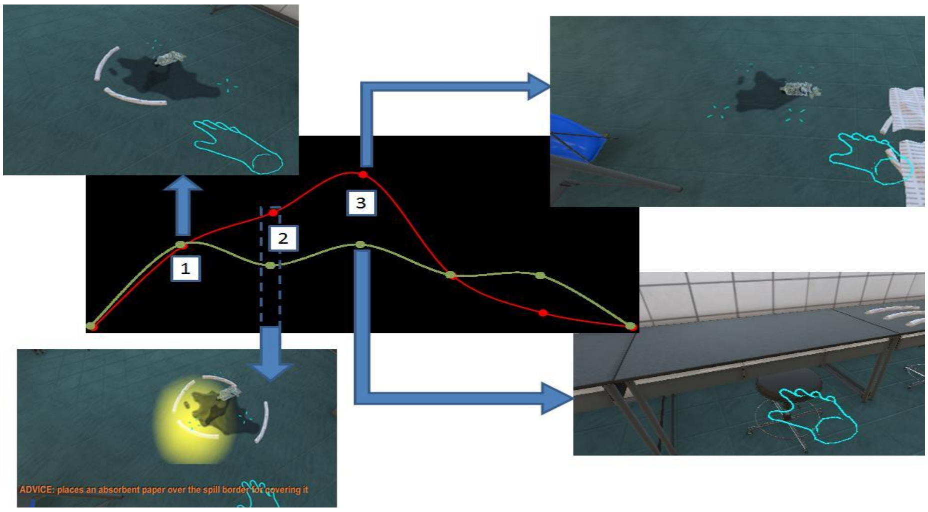

1
2
3
ADVICE: places an absorbent paper over the spill border for covering it

Fig. 2. Different behaviors of a scenario when using two different Intensity curves.

Figure 2 shows an example of how two different curves would affect the curse of the events in one hypothetical training scenario creating two different sessions as a result (given by the red and green curves; we will refer to the two different generated sessions as red and green respectively). Given that the user acts exactly the same way in both sessions, the events until the first milestone (Number 1 in the figure) of the curves would be roughly the same, raising the Intensity of the session until intermediate values (represented by triggering a spill overflow, pictured in the image). Then the two curves separate and the red one starts to rise whilst the green one decreases. In the red session this represent that the user likely will receive more high impact events, allowing a fast increase in the Intensity. In the green one, the Narrative Manager will trigger only some low impact events, lowering the Intensity. After the session reaches the second milestone (Number 2), both curves rises, so the Narrative Manager decides to give a hint to the user in both sessions, but given that the next milestone in the green curve is lower, probably it will be reached sooner than in the red session. In that one the Intensity will increase over the time at a faster rate and also will trigger more events in order to reach the higher milestone. The effect of the different Intensity values being bounded by the curve can be seen clearly in the third milestone of the curves (Number 3), where the red one has a high value and allows the system to trigger a new spill (an event with very high impact). On the other hand, the green curve will trigger a more moderate event, like hiding one of the spill cleaning tools. As we see, two different curves would condition how the Narrative Manager select the events (even with exactly the same user actions) by limiting the maximum impact that the selected events can have and by modifying the rate at the Intensity rises.

5 Conclusion

In this paper, we have presented an authoring tool designed to control the way of a Narrative Manager behaves in a virtual training application. In our system, the Narrative Manager decides which event to trigger attending to a group of parameters and with the goal of improving the user experience from the simulation. The authoring tool we developed enables an intuitive method to limiting the dramatic intensity that the session will have at any moment by generating a model we called Intensity Curve and using it as an input for the Narrative Manager. Also, our tool has an abstraction level high enough to not being necessary to specify the details of the events for the training sessions and give the Narrative Manager freedom to decide them, thus creating different sessions each time the training is run, even with the same Intensity Curve.

Our first experiments with the Intensity Curve Modeler application have allowed us to validate the architecture and the requirements of the authoring tool: when the curve has low values the system does not generate high impact events, maintaining a low intensity, and on the contrary, when the curve has a peak the system will generate a high impact event for the user. However, the algorithm used for our authoring tool when the Narrative manager is fed with an Intensity Curve is one interpretation from a number of possible ones, the focus of this works is to design and develop a functional and effective way for authoring the narrative of the session. Nevertheless, more testing is needed, and we plan to perform an experiment in real conditions with real subjects and for comparing with other strategies for drama modeling. Also, there are other aspects we want to improve in the authoring tool: first, we want to add a feature for input event preferences for the session: we stated that the training designer may want to control the intensity of the session, but maybe he also may want to focus the training in some type of events in concrete. And second, we want to allow creating certain rules for the user performance: the scenario designer may want to specify different ways for the session to behave depending of the user performance. It could be argued that by allowing these features in the authoring tool we are increasing again the tasks of the scenario designer, something opposite to the goals of including the Narrative Manager so it is very important to keep a balance between customizing capability and automatic dynamic generation. We think that researching and experimenting deeper on these issues is the way to provide better authoring tools in the complex field of interactive virtual environments, and a necessity for virtual training applications.

References

1. Rizzo, A., Parsons, T., Buckwalter, G. and Kenny, P. (2010). A New Generation of Intelligent Virtual Patients for Clinical Training. In IEEE Virtual Reality Conference, Waltham, USA.
2. Reger, G. M., Holloway, K. M., Candy, C., Rothbaum, B. O., Difede, J., Rizzo, A. A. and Gahm, G. A. (2011), Effectiveness of virtual reality exposure therapy for active duty soldiers in a military mental health clinic. J. Traum. Stress, 24: 93–96. doi: 10.1002/jts.20574.
3. Raybourn, E. M., Deagle, E., Mendina, K., & Heneghan, J. (2005). Adaptive thinking & leadership simulation game training for special forces officers. Proceedings of Interservice/Industry Training, Simulation, and Education Conference. Orlando, FL.
4. Carpenter, E., Kim, I., Arns, L. L., Dutta-Berman, M. J., & Madhavan, K. P. C. (2006). Developing a 3D simulated bio-terror crises communication training module. In VRST, M. Slater, Y. Kitamura, A. Tal, A. Amditis, and Y. Chrysanthou, Eds., ACM, 342–345.
5. Gordon, A. S. (2004). Authoring branching storylines for training applications. In *Proceedings of the 6th international conference on Learning sciences* (pp. 230-237). International Society of the Learning Sciences.
6. Ponder, M., Herbelin, B., Molet, T., Schertenlieb, S., Ulicny, B., Papagiannakis, G., Magnenat-Thalmann, N., & Thalmann, D. (2003). Immersive VR decision training: Telling interactive stories featuring advanced virtual human simulation technologies. In 7th International Workshop on Immersive Projection Technology, 9th Eurographics Workshop on Virtual Enviroments, A. Kunz and J. Deisinger, Eds., Eurographics Association (Zurich, Switzerland), 097–106.
7. Silverman, B. G., Johns, M., Weaver, R. & Mosley, J. (2003). Authoring Edutainment Stories for Online Players (AESOP): Introducing Gameplay into Interactive Dramas. In O. Balet, G. Subsol & P. Torguet (ed.), International Conference on Virtual Storytelling, Vol. 2897 (pp. 65-73). Springer. ISBN: 3-540-20535-7.
8. Mollet, N., Arnaldi, B. (2006). Storytelling in Virtual Reality for Training. Edutainment 2006:334-347.
9. Habonneau, N., Richle, U., Szilas, N. & Dumas J. E. (2012). 3D Simulated Interactive Drama for Teenagers Coping with a Traumatic Brain Injury in a Parent. In LCNS Interactive Storytelling. Springer Berlin Heidelberg.
10. Sottilare, R. A., & Holden, H. K. (2013, July). Motivations for a Generalized Intelligent Framework for Tutoring (GIFT) for Authoring, Instruction and Analysis. In AIED 2013 Workshops Proceedings Volume 7 (p. 1).
11. Floryan, M., & Woolf, B. P. (2013, January). Authoring Expert Knowledge Bases for Intelligent Tutors through Crowdsourcing. In *Artificial Intelligence in Education* (pp. 640-643). Springer Berlin Heidelberg.
12. Tripoliti, E.E.; Pappas, I.G.; Petrakis, E.G.M.; Sans, J.M., "Crafting vascular medicine training scenarios: The RT3S authoring tool," Bioinformatics and Bioengineering (BIBE), 2013 IEEE 13th International Conference on , vol., no., pp.1,4, 10-13 Nov. 2013
13. Shaker, N.; Shaker, M.; Togelius, J.. Ropossum: An Authoring Tool for Designing, Optimizing and Solving Cut the Rope Levels. AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, North America, nov. 2013.
14. Isotani, S., Mizoguchi, R., Isotani, S., Capeli, O. M., Isotani, N., & de Albuquerque, A. R. (2010, January). An authoring tool to support the design and use of theory-based collaborative learning activities. In Intelligent Tutoring Systems (pp. 92-102). Springer Berlin Heidelberg.

15. Karampiperis, P., Mouchakis, G., Paliouras, G. and Karkaletsis, V. 2011. ER designer toolkit: a graphical event definition authoring tool. In Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments (PETRA '11). ACM, New York, NY, USA, , Article 34 , 5 pages.
16. Cablé, B., Guin, N., & Lefevre, M. (2013). An Authoring Tool for Semi-automatic Generation of Self-assessment Exercises. In Artificial Intelligence in Education (pp. 679-682). Springer Berlin Heidelberg.
17. Olsen, J. K., Belenky, D. M., Aleven, V., Rummel, N., & Ringenberg, M. (2013, June). 1 Authoring Collaborative Intelligent Tutoring Systems. In AIED 2013 Workshops Proceedings (p. 1).
18. Sottilare, R.A., Brawner, K.W., Goldberg, B.S., & and Holden, H.K. (2012). The Generalized Intelligent Framework for Tutoring (GIFT). Orlando, FL: U.S. Army Research Laboratory – Human Research & Engineering Directorate (ARL-HRED).
19. Amin-Naseri, M., Guo, E., Gilbert, S., Jackman, J., Hagge, M., Starns, G., & Faidly, L. (2013, July). Authoring a Thermodynamics Cycle Tutor Using GIFT. In AIED 2013 Workshops Proceedings Volume 7 (p. 45).
20. Ray, C., & Gilbert, S. (2013, July). Bringing Authoring Tools for Intelligent Tutoring Systems and Serious Games Closer Together: Integrating GIFT with the Unity Game Engine. In AIED 2013 Workshops Proceedings Volume 7 (p. 37).
21. Aleven, V., Sewall, J., McLaren, B. M., & Koedinger, K. R. (2006, July). Rapid authoring of intelligent tutors for real-world and experimental use. In Advanced Learning Technologies, 2006. Sixth International Conference on (pp. 847-851). IEEE.
22. Paquette, L., Lebeau, J. F., & Mayers, A. (2010). Authoring Problem-Solving Tutors: A Comparison between ASTUS and CTAT. In Advances in Intelligent Tutoring Systems (pp. 377-405). Springer Berlin Heidelberg.
23. Olsen, J. K., Belenky, D. M., Aleven, V., & Rummel, N. (2013, January). Intelligent Tutoring Systems for Collaborative Learning: Enhancements to Authoring Tools. In Artificial Intelligence in Education (pp. 900-903). Springer Berlin Heidelberg.
24. Jee, H. K., Lim, S., Youn, J., & Lee, J. (2014). An augmented reality-based authoring tool for E-learning applications. Multimedia Tools and Applications, 68(2), 225-235.
25. Alvarez, N., Prendinger, H., Cavazza, M. Shigematsu, M., Sanchez, A., "Narrative Balance Management in an Intelligent Biosafety Training Application for Improving User Interest." Accepted with minor revisions in *International Journal of Artificial Intelligence in Education* (2014).
26. Ware, S. G. & Young, R. M. (2011). Toward a Computational Model of Narrative Conflict. Technical Report DGRC-2011-01, Digital Games Research Center, North Carolina State University, Raleigh, NC, USA. http://dgrc.ncsu.edu/pubs/dgrc-2011-01.pdf.