https://doi.org/10.2352/ISSN.2470-1173.2018.03.ERVR-469

An authoring system for VR-based firefighting commanders training

Diego Puel, Delta Informatica, Trento, Italy – Paolo Busetta, Delta Informatica, Trento, Italy – Nicola Conci, University of Trento, Italy

Abstract

Building a realistic training environment to teach situation awareness and coordination of teams implies more than good graphics and HCI. Plausible and varied behavior of the artificial characters, richness and variety of the stories experienced by the trainee are essential features of an effective training session. This paper discusses an authoring environment for simulations targeted at the training of firefighting commanders. This environment, called PRESTO-FF, currently includes a modelling facility for behaviours of people on the scene, a specific 3D environment, and a visual scripting system that drives the unfolding of the simulation according to the player's actions and real-time choices by a trainer. Models and scripts are reusable in multiple 3D environments thanks to abstractions and semantics provided by underlying AI technologies. Results of the validation of a simulation with domain experts are reported, highlighting both advantages and limitations of the current framework.

Introduction

Activities exposed to high safety risks, such as firefighting and, more in general, rescue operations, can benefit from computer-based simulations and VR to safely experience complex and hazardous situations. Such contexts, where the dynamic of the ongoing event is rapidly evolving, imply taking rapid decisions, for which an appropriate training of the personnel is required. The use of simulation is not new per se; indeed, it has seen major application in the military domain. In fact, it is particularly relevant for coordination roles, which need to learn how to deal with uncertainty and to anticipate risks that may involve teammates, victims and passersby. Due to the complexity and the high variability of such events, which also expose personnel operating on the field to hazardous situations, it is difficult to reproduce the scene on a real-life training field (e.g. large fires in crowded buildings). So-called VR-based "serious games" can then be a potential solution. Concerning firefighting, commercial systems such as FLAME-SIM (http://www.flame-sim.com/), Designing-Digitally's Firefighter Training (http://www.designingdigitally.com/), and ADMS-Fire (http://www.trainingfordisastermanagement.com/products/adms-fire) focus on the individual firefighter's operations needed to maneuver equipment or learn intervention procedures; when team work and coordination are supported, they require multiple players involved in various roles. This state of the art partially reflects the current limitations of so-called artificial intelligence for games, especially concerning behaviors of characters, which is key for the training of coordinators.

The work presented in this paper builds on an infrastructure for the development of serious games with support for behavioral modelling and story scripting, PRESTO [1], to provide an environment for an instructional designer to rapidly build complex firefighting scenarios involving many people with different roles. This environment, called PRESTO-FF (PRESTO FireFigthing), allows the creation of immersive simulations for those in charge of coordinating multiple squads in an accident (i.e. Incident Commanders in the US terminology, ROS in Italian), providing a first-person view of the location in a similar way to what he would experience in real-life. This environment includes behavioral models of teammates and other people involved, scripts controlling physical phenomena (e.g., how the fire evolves and spreads), scripts determining the evolution of the overall situation according to occurring events, contributing to the creation of highly dynamic scenarios. The simulations are supervised by a trainer, who selects the scripts to be run and can influence the unfolding of the story at specifically designed times.

Considering that the training focus is on awareness and strategic decision-making, we are not addressing here additional elements as equipment usage or team tactics, which are of undisputable interest but are not directly a matter for the operation coordinator.

To validate the proposed solution, a case study has been built and validated with a team of experts. The achieved results are encouraging, and the received feedback will be essential to enrich the framework with additional features.

The approach presented here is of interest to the VR engineering community both for its supervised approach to training, and because of how it addresses one of the most critical limitation of the currently available systems, which prevents the large-scale adoption of VR, namely rapid and cost-effective development of scenarios, no matter whether for serious or entertainment purposes.

The rest of this paper is structured as follows: next section briefly introduces the main modality of use of PRESTO-FF. The following section is a short overview of the 3D environment and the abstraction and semantic layers provided by PRESTO. Then, the so-called artificial intelligence (AI for short) in use in PRESTO-FF to model behaviors, as well as its player-to-AI interaction mechanism, are briefly discussed. The scripting environment that represents the authoring facility of PRESTO-FF, exploiting available semantics and behavioral AI, is described. The case study built with PRESTO-FF is discussed, reporting the results of validation with domain experts. In conclusion, a few possible directions for future developments and research are outlined.

Simulation as e-learning classroom tool

As mentioned in the introduction, simulations built with PRESTO and PRESTO-FF are typically used in a classroom setting. The first-person view of the player is projected on a large screen. A student or a trainer controls movement and actions via a joystick, while the others look and can suggest actions. The trainer observes the students and can comment what is happening in the VR, e.g. to explain the situation or suggest choices. The trainer has a secondary screen from which s/he can control which scripts to start and can take choices suggested by those running, according to the current training objectives and students' reactions. Also, the trainer can rollback the simulation to predefined points in time, again determined by running scripts, thus allowing to re-experience events and possibly take different actions.

This is not the only possible configuration, but is very appreciated by trainers because it gives them the opportunity to discuss with the classroom, to highlight specific issues and to exploit the flexibility offered by properly designed scripts to achieve variety, a high level of interest by students and to exercise their awareness and decision-making skills.

3D environment and semantic facilities

PRESTO uses a commercial product, XVR [2], which allows to rapidly build a 3D scenario by selecting one of the available prebuilt 3D environments and populate it with objects, characters and predefined animations from XVR's ample 3D library. The scenarios are used to run interactive simulations in XVR's so-called "training mode". XVR supports one or more players, each having the first-person view of a character s/he controls, plus a trainer's station that allows to manipulate the current scenario in real-time. Observe that the typical PRESTO-FF classroom setting described above is single player for training reasons, not for technical limitations.

PRESTO uses ontologies [3] to give meaning to locations (e.g. in firefighting, "fire location", "fire exit"), independently of their coordinates and graphical representation, as well as to classify all items on the scene and to dynamically annotate them with data that can be used for reasoning (so-called "qualities"), including states, actions being performed, relationships with other items. This allows models and scripts to be independent of a specific 3D environment. Further, semantic classification and annotations enable the definition of script situations, described later.

The current version of PRESTO-FF adopts one specific XVR environment that includes an apartment building and its neighborhood; a labor-intensive but technically trivial future extension is the selection and semantic annotation of new environments.

Modelling behaviors. Interacting with the AI

PRESTO's original framework for NPC (Non Player Character, e.g. an avatar representing a human) behavioral modelling is called DICE [4]. DICE implements extensions to the well-known BDI (Belief Desire Intention) agent architecture [6] directed at making models modular, easy to compose and integrated with the semantic facilities of PRESTO. BDI prescribes a clear differentiation between goals (or Desires), i.e. "what" an agent has to achieve, and plans (or Intentions), i.e. "how" an agent intends to achieve its goals in the current context. DICE introduces various extensions to BDI for meta-reasoning (i.e. reasoning about what the agent is doing), which can be exploited for various purposes including coordination [5]. Further, it uses meta-data to represent an agent's roles (i.e. the sets of goals it can achieve) and behavioral models available to the agent (i.e. the libraries of procedures that achieve the goals of the roles supported by the agent). The behavioral model for a role of the agent can be dynamically chosen depending e.g. on the NPC's state, thus allowing a modeler to differentiate models according to predetermined skills and personality traits and to evolving parameter values such as fatigue. Further, DICE adds other parameters to BDI inspired by a cognitive architecture called CoJACK [7].

PRESTO exploits the meta-data of roles to introduce a form of direct player-NPC interaction that extends the native HCI of the VR engine in use (XVR or Unity): a goal can be interactively submitted by selecting the character and choosing one of its goals that have been appropriately annotated on the roles' meta-data (which in turn is dynamically modifiable via an API).

In addition to a set of roles and behavioral models specific for firefighting scenarios, some of which directed at coordinated work within a squad, PRESTO-FF introduces two major modelling innovations to DICE and PRESTO. The first is strictly technical: a novel BDI framework based on C# co-routines, aiming at faster model development and better performance than the original DICE implementation thanks to the use of state-of-the-art programming tools. The second consists in the introduction of a pseudo-NPC, called "radio", which acts as a virtual assistant to the player. The radio's goals can be submitted interactively by the player at any time, as if the player were talking to an (invisible) radio indeed. The radio model should be developed according to the simulation's purposes; in the case of PRESTO-FF, it allows to submit certain goals to the firefighters and the squads present in the scene (based also on their current activities), and can query all team member's current states (i.e. the main goals they are pursuing) to report them to the player, simulating silence by those that communicate of being victims of a deadly accident. The radio model is then a powerful and flexible mechanism for extending the VR's HCI in a context-sensitive and simulation-specific way.

Visual scripting and run-time control

The key authoring tool offered by PRESTO, fully exploited by PRESTO-FF, is its scripting language, PRESTO Script [8], with its accompanying visual editor, engine and control GUI. PRESTO Script allows to describe stories, situations, scenes and consequences through graphs, drawing a choreography of entity behaviours within the simulation. It allows to have an arbitrary number of scripts that run in parallel, controlling the development of the story by means of unfolding "scenes" in which NPCs are given goals to pursue while entities in the VR are manipulated by changing their qualities.

A script is a cyclic directed graph, where nodes represent scenes and edges represent events that push the story ahead, possibly along parallel paths; arbitrary paths representing alternative stories can be described. Scenes may be marked as "checkpoints", i.e. important points in the story to which the trainer can go back during a simulation. Events may include interactive choices by the trainer (e.g. "continue with story 1" vs "continue with story 2") or may represent the matching of the state of the VR (including the activities of the NPCs) against arbitrary patterns called "situations". A situation may capture a complex combination such as "there is a fire in a room with people choking from smoke and a firefighter opening the door". Situation detection is performed at run-time thanks to the underlying PRESTO semantic facilities mentioned earlier.

Scripts and situation definitions are edited by means of an ad-hoc visual editor, meant for use by domain experts defining their simulations based on the available 3D and behaviour models.

As mentioned earlier, scripts are started and interactive choices taken by means of a GUI (called PRESTO Script controller) running on a different screen than the VR, thus the trainer does not interfere with the player.

A simulator: case study and validation

The validation of PRESTO-FF has been conducted by subjective evaluation, involving experts in firefighters training. The objective is to determine the suitability of the system as a training environment for a ROS (Italian for Incident Commander). For this purpose, an ad-hoc scenario with a number of relevant scripts has been developed with PRESTO-FF and the outcome of the validation is analyzed through a questionnaire aimed at assessing different features of the simulation environment ranging from the behavioral models to the usability of the interface. All the relevant details are provided in the next paragraphs.

Case Study

We decided to set up an environment addressing some relevant objectives that a potential trainer could use for building a specific course or as integration of an existing one. The trainer is meant to be the controller of the simulation, playing the role of the ROS and guiding the audience in the classroom settings described earlier. As the variability of the possible scenarios is very high, we focused on a limited but highly representative set of training objectives in presence of a fire in an apartment, including: analysis of the scene; positioning of rescue vehicles; management of a crowd of passers-by; radio communication with team members; squad management for fire extinction, search and rescue, ventilation; back-draft (i.e. an explosion caused by inflow of air following the opening of a door) and gas explosions during intervention; dealing with casualties (both civil and squad members).
The modeling of the above-mentioned objectives has driven the definition of roles and goals of the NPCs required to dynamically handle the evolution of the scene. The main roles and the corresponding goals are briefly reported in Table 1; their names, built with a simplified English syntax, are meant to match the terminology and expected meaning by domain experts.

Table 1. Main roles and their supported goals

ROLE NAME	GOAL NAMES
Firefigther	ExtinguishFire, ExtinguishAllFires, ConnectHose, SearchVictims, SaveVictim, SearchFirefigtherVictim, SaveFirefigther, GiveInfo
ForemanExtinguish	StartExtinguish, StartVentilate, SquadExtinguishFire, SquadVentilate
ForemanSearch	StartSearch, StartSearchFirefigthers, SquadSearch, SquadSearchFirefigthers
Informer	CatchAttention, DescribeEvent, FollowROS
Victim	BeVictim, StayAway
CrowdMember	BeCrowd, MakeHero
Radio	SetAPSPosition, SetLadderPosition, Foreman_<number>_Extinguish, Foreman_<number>_Search, Foreman_<number>_SearchFirefigthers, Foreman_<number>_Ventilate, CallAnotherFireBrigade

To this end, however, certain name conventions need to be enforced; for instance, the goals prefixed with "Squad" in the "ForemanExtinguish" role are those meant for a Foreman as squad leader coordinating two other firefighters for extinguishing a fire (goals typically submitted by the ROS via radio) rather than the specific operations as a firefighter her/himself (prefixed with "Start" in this case).
The radio goals, as mentioned above, correspond to commands that the player can give to other NPCs via radio communication. These roles are meant to be reusable in multiple environments and potentially be assigned to different types of agents. The agents employed in the case study are listed in Table 2.
As can be seen from Table 2, also the variability of the agents is high, and each of them can be assigned different roles driven by the corresponding behavioral model. The variability in the agent behavior clearly depends on its complexity; agents related to human behaviors are likely to have different facets, while objects (vehicle, radio) are expected to be represented by single specific models.

Table 2. Main agents developed for the case study, their main roles and the associated behavioral models.

AGENT	MAIN ROLE	BEHAVIORAL MODEL
Crowd	CrowdMember	NormalCrowd
Firefighter	Firefighter	NormalFirefighter
	Dead	Dead
Foreman	ForemanExtinguish	NormalExtinguishForeman
	ForemanSearch	NormalSearchForeman
InvolvedPerson	Informer	AgitatedInformer
Radio	Radio	Radio
Victim	Victim	NormalVictim

Framework validation

Relying on the agents and roles described in Table 1 and Table 2, fifteen domain experts, including chiefs, platoon leaders and foremen as well as trainers, were asked to evaluate the simulation framework looking at a predefined emergency scenario addressing specific training objectives. This represents the development of an apartment fire and the consequent rescuer intervention (Figure 1). It encompasses a number of objectives, including the assessment of the incident scene, detecting the presence of the crowd, locating the fire zone, identifying where to position the intervention vehicles and reasoning about the intervention tactics to use, as well as the presence of informers (who might report correct or wrong information). The ROS can assign orders to the squad via radio communication, as fire extinguishing and victims search (Figure 2 and Figure 3). He can face the presence of anomalous events, like a person rushing in the building, casualties, chaos and crowds. All these events are not known a priori by the trainee and have then to be addressed on the spot. A video with an overview of the authoring tools and an example of a session from the case study is available on YouTube (https://youtu.be/5LoncV6J3Os).
The validation session consisted of a first introductory phase, during which the project and the platform environment together with the tools available to the trainer were presented. After a complete live simulation was run, the experts were asked to fill the questionnaire.
The questionnaire has been arranged in 8 separate sections, dealing with the reliability of the behavioral models and the generated events, the perceived usability and flexibility of the framework, as well as general impressions about the graphical layout. A summary of the qualitative evaluation of the experts is provided for 5 out of 8 sections in Table 3. Results are expressed in terms of average score, in a range from 1 to 4, where 4 is the maximum appreciation (based on a Likert scale).

Figure 1. Incident case, apartment fire.

Figure 2. Goal outcome of ExtinguishFire.

Figure 3. Goal outcome of SearchVictims

Table 3. Outcome of the evaluation questionnaire.

Parameter	Average Score
Graphical layout	2,47
Plausibility of events	2,91
Plausibility of behaviors	2,9
Flexibility	3,44
Usability	3,63
Would use as standalone game without trainer	3,02

Conclusions

PRESTO-FF is a novel authoring environment for firefighting commander training. It provides an environment for instructional designer to describe potential stories that are chosen and controlled by a trainer during a simulation with trainees. Validation of a case study has shown a high level of appreciation in spite of the current graphics considered below expectations. Future works will mostly focus on improved behavioral modelling, introducing team facilities for coordinated behavior as well as psychological factors that have been only marginally considered in the current version.

References

[1] M. Robol, P. Calanca, P. Busetta, "Applying BDI To Serious Games: The PRESTO Experience", University of Trento, Technical Report DISI-16- 011, December 2016, Available at http://hdl.handle.net/11572/154697

[2] XVR Simulations BV, "XVR On Scene" [Online]. Available at http://www.xvrsim.com/en/XVR_Platform/XVR_On_Scene/.

[3] M. Dragoni, C. Ghidini, P. Busetta, M. Fruet, M. Pedrotti, "Using Ontologies For Modeling Virtual Reality Scenarios", Proceedings of ESWC 2015, June 2015 – Springer, 2015

[4] P. Busetta, M. Dragoni, "Composing Cognitive Agents from Behavioural Models in PRESTO", [Online] Proceedings of WOA 2015 workshop, Napoli (Italy), June 2015. Available at http://ceur-ws.org/Vol-1382/paper13.pdf

[5] M. Robol, P. Giorgini, P. Busetta, "Applying social norms to high-fidelity pedestrian and traffic simulations", Proceedings of ISC2 2016, Trento (Italy), Sep. 2016

[6] A. S. Rao, M. P. Georgeff, "Bdi agents: From theory to practice", in Proceedings of The First International Conference on Multi-Agent Systems (ICMAS-95, pp. 312– 319, (1995).

[7] F. E. Ritter, J. L. Bittner, S. E. Kase, R. Evertsz, M. Pedrotti, P. Busetta, "CoJACK: A high-level cognitive archi- tecture with demonstrations of moderators, variability, and implications for situation awareness", Biologically Inspired Cognitive Architectures, 1, 2–13, (July 2012).

[8] P. Busetta, M. Robol, P. Calanca, P. Giorgini, "PRESTO Script: scripting for serious games", AI & Games Symposium at AISB 2017, University of Bath (UK), 18/22 Aprile 2017

Authors Biography

Diego Puel received the Master degree in Telecommunication Engineering at the University of Trento, Italy. He currently works as network engineer for Delta Informatica. This paper describes his work for its Master thesis, concluded in July 2017.

Paolo Busetta is the technical head of the R&D department of Delta Informatica, where he led the developments of PRESTO. His research work has mainly focused on agents and simulation. Paolo has a Master degree from the University of Melbourne, Australia (1999) and a degree in Computer Science (Laurea) from the University of Torino, Italy (1986).

Nicola Conci is Assistant Professor at the University of Trento, Italy. He was a Visiting Student with the Image Processing Lab., UCSB, CA, USA, and a Postdoctoral Researcher with the Multimedia and Vision Research Group, Queen Mary University of London, U.K. He has authored and coauthored more than 80 scientific papers in international journals and conferences. He is cofounder of the startup Xtensa S.r.l. His current research is in the area of video analysis for human behavior analysis with particular application to environmental monitoring, surveillance, and assisted living.