Training Support for Crisis Managers with Elements of Serious Gaming

Denis Havlik, Oren Deri, Kalev Rannat, Manuel Warum, Chaim Rafalowski, Kuldar Taveter, Peter Kutschera, Merik Meriste

Training Support for Crisis Managers with Elements of Serious Gaming

Denis Havlik[1], Oren Deri, Kalev Rannat, Manuel Warum, Chaim Rafalowski, Kuldar Taveter, Peter Kutschera, Merik Meriste

[1] Austrian Institute of Technology, Vienna, Austria
Denis.havlik@ait.ac.at

Abstract. This paper presents a methodology and a prototypic software implementation of a simple system supporting resource management training for crisis managers. The application that is presented supports the execution and assessment of a desktop training for decision makers on a tactical and strategic level. It introduces elements of turn-based strategic “serious gaming”, with a possibility to roll back in time and re-try new decision paths, while keeping the graphical user interface as simple as possible. Consequently, the development efforts concentrated on: (1) formulating and executing crisis management decisions; (2) assuring responses of all simulated entities adhere to natural laws of the real world; and (3) analyzing progress and final results of the training exercise. The paper presents the lessons learned and discusses the transferability and extensibility of the proposed solution beyond the initial scenario involving accidental release of toxic gas in an urban area in Israel.

Keywords: crisis management · training · resource management

1 Introduction

Organizing realistic trainings which force crisis managers and other training participants to understand that “there is no ideal solution to a complex crisis” is difficult [1]. Trainers, trainees and other stakeholders often oversee the obvious mistakes in both, planning of the (field) trainings and assessing the training results. Typical omissions include: the unlimited availability of resources, unrealistically fast arrival of resources at the scene, instantaneous and unfailable performing of some tasks such as a triage, as well as the superhuman endurances of the responders and patients alike.

The CRISMA project team developed a methodology and a set of software tools which can be used to make trainings quite realistic, by keeping most of the training organization unchanged and, where needed, taking into account the peculiarities of the local crisis management organization.

Below we start with a presentation of one concrete use case targeted in the “Accidental pollution” application of the CRISMA project. Thereafter, in section 3, we introduce the technical implementation of the application and section 4 presents the

adfa, p. 1, 2011.

lessons learned in application development and testing, and close with the discussion on transferability and extensibility of the proposed solution.

2 Decision Support Concept and the Resource Management Training Scenario

The CRISMA project is a European Union funded project focusing on simulation of multi-sectorial large scale crisis scenarios with multi-dimensional effects on the society. According to the CRISMA architecture [2,3] the aspects of the *World* we are interested in must be modelled as a set of representative *World State* data (*WSs*) which develops in discrete steps (*World Transitions*), depending on the inherent characteristics of the problem at hand as well as on the users' decisions (*Decision Chain*).

Depending on the selected use case, new WSs may be generated at regular time intervals, following the users decisions (e.g. "evacuate"), or to capture the results of a specific model execution (e.g. "ambulance has arrived to the scene"). In most CRISMA applications, the user can test the effects of alternative decisions. This results in distinct *Scenarios, i.e.* sets of WSs corresponding to unique *Decision Chains*.

WSs and whole Scenarios can be compared and assessed using a set of *Indicators* relevant to a problem at hand. In addition to indicators, CRISMA also supports *Criteria* – a normalized form of the Indicators which take into account the desired values of the Indicators (e.g. "evacuating the critical patients in less than 30 min is *good*"). This simplifies the task of analyzing the results and ranking the different Scenarios. However, the Criteria are both subjective and extremely sensitive to small changes in the problem definition. Consequently, the criteria must be defined separately for each training or planning session, e.g. as a part of the training setup.

The CRISMA decision support concept can thus be summarized as: *help the users to assess the results of their decisions, but do not suggest a course of action nor impose any particular solutions.*

2.1 Resource Management Training Scenario

The CRISMA concepts and the software framework have been tested in five pilot applications covering various aspects of the crisis management planning and training [4]. Two of these applications target the long-term infrastructure and organizational planning, one concentrates on the development of operational plans, and the final two address the planning, execution and assessment of the resource management trainings. Here and now we only concentrate on the resource management training application.

During an emergency, commanders in emergency organizations have to decide which resources (ambulances, fireman, police officers, etc.) to deploy, where to send them, and what their tasks should be. In our trainings, the commanders (trainees) are confronted with initial crisis situation and have to express their decisions as they would do during a real accident. Thereafter, the decisions are entered into the system by a steward, the application calculates a new situation, and the result is presented to the trainees. At every step, the trainee can issue additional commands or even step backwards and revise previous decisions (**Fig. 1**).

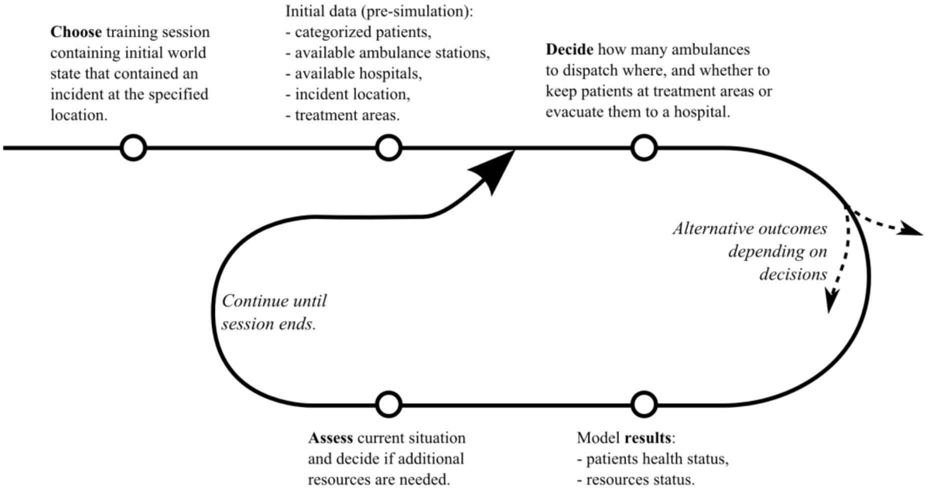

Choose training session containing initial world state that contained an incident at the specified location.
Initial data (pre-simulation):
- categorized patients,
- available ambulance stations,
- available hospitals,
- incident location,
- treatment areas.
Decide how many ambulances to dispatch where, and whether to keep patients at treatment areas or evacuate them to a hospital.
Alternative outcomes depending on decisions
Continue until session ends.
Assess current situation and decide if additional resources are needed.
Model **results**:
- patients health status,
- resources status.

Fig. 1. Resource management training workflow

A concrete accident scenario addressed in the resource management training pilot deals with an accidental spilling of chemicals (i.e. bromine) from a container in a major port. The resulting plume poses a threat to a large number of inhabitants in the neighboring city and the severity of the impact depends on the commander's decisions. The main purpose of the application is to assure the impact of the decisions taken by the trainees is realistic, in a sense that the impact is guided by the natural laws and peculiarities of the training setup. For example:

- Resources have to be actively deployed and their arrival to the scene is delayed based on their distance from the scene, type of the road, and weather conditions.
- The condition of the patient will deteriorate based of one's previous medical history, exposure to the contamination and the help received (e.g. decontamination).
- The condition of resources also changes as result of the interaction between the resource and the environment and/or patient. At some point, resources may become exhausted and cannot perform additional actions.

3 Concept and Implementation

A CRISMA resource management training application is realized by using six generic "Building Blocks" (BBs) of the CRISMA framework (**Fig. 2**).

- *Integrated Crisis Management Middleware (ICMM)* is the core CRISMA middleware service. It acts as a catalogue of the WSs and Scenarios and also supports the work with Indicators and Criteria. *Objects of Interest World State Repository*

(OOI-WS Repository) and the *Agents Platform* assure that resources of interest for the application are represented in a common form and that their state changes over time, similarly to a way it would be in 'real life'. Main OOIs and activities modeled in the pilot are illustrated in the **Fig. 3**.

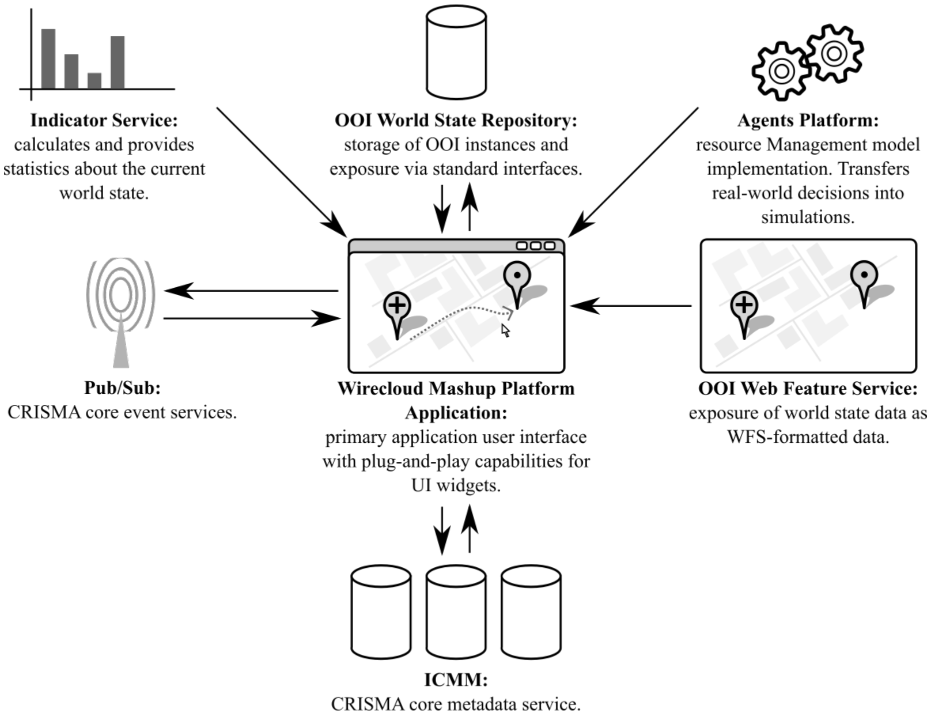

Indicator Service:
calculates and provides statistics about the current world state.
OOI World State Repository:
storage of OOI instances and exposure via standard interfaces.
Agents Platform:
resource Management model implementation. Transfers real-world decisions into simulations.
Pub/Sub:
CRISMA core event services.
Wirecloud Mashup Platform Application:
primary application user interface with plug-and-play capabilities for UI widgets.
OOI Web Feature Service:
exposure of world state data as WFS-formatted data.
ICMM:
CRISMA core metadata service.

Fig. 2. Functional elements of the resource management training application

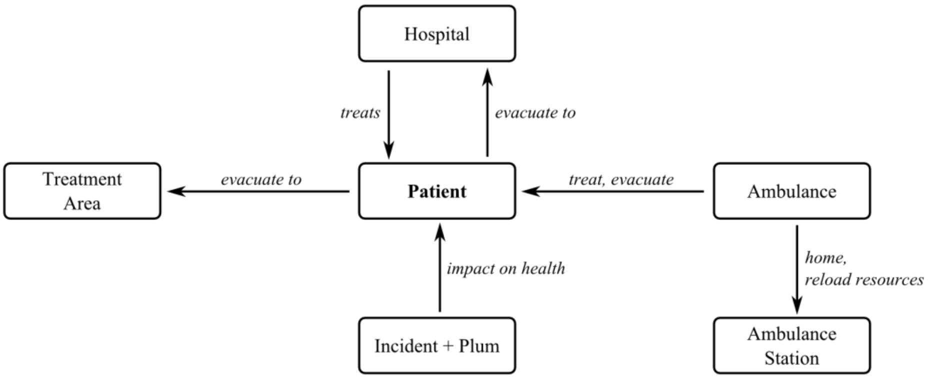


Fig. 3. Main Objects of Interest and activities

- *Indicator Service* calculates the relevant indicators whenever a new World State has been generated by the user or by the agents' platform. Indicators (e.g. "number of dead", "utilization of resources" or "time required for evacuation of critical patients") are used to analyze the progress and final outcome of the training.
- *Pub/Sub service* provides support for event-driven application designs. In this application, all changes to the WS are advertised using the Pub/Sub service. Software elements such as the agents' platform or the indicator service are subscribed to these events and automatically activated as needed.
- Finally, the *Wirecloud Mashup Platform* provides a flexible platform for implementing the application GUI elements and building mashup applications.

The following sections provide more information on three central system components: OOI-WS repository, Agents platform and the Mashup application.

3.1 Objects of Interest -World State Repository

OOI-WSR is a service which enables archiving, querying and manipulation the Objects of Interest world state data. It provides a central information exchange and data storage point for all other BBs which consume and manipulate world state data.

The repository data is organized in simulation sessions. Each repository session corresponds to a distinct training and consists of multiple world state snapshots. Each of the snapshots represents the state of the world at specific points (in time) and contains the states of all OOIs relevant for the training.

The starting properties of all OOIs in the initial world state (e.g. initial positions and states of citizens and ambulances) are postulated by a trainer. Subsequent world state snapshots are generated by the agents' platform BB and reflect both the inherent model logic and the trainees' decisions.

The OOI-WSR provides an easy http based access to data and enables data manipulations for authorized services. It exposes the stored data in a variety of industry standard such as REST/JSON and OGC WFS/GML in order to support out-of-the-box integration with other web technologies. The repository assures that each OOI is an instance of a specific OOI type representing some real-world resource (e.g. "ambulance"). In order to facilitate data analysis, typed OOIs also prevent the trainers from instantiating the "impossible" resources as a part of the training setup.

3.2 OOI Models and Serious Gaming

The general logic and architecture of CRISMA favor the development of strategic turn-based "serious gaming" applications. This type of applications emphasizes making strategic and/or tactical decisions by the players, rather than elaborate graphics or real time reaction skills. A central concept in serious gaming is that of an "agent", which is understood as a proactive and social entity capable of reasoning [1,5,6,7,8].

In our training application, each of the OOI types defined in the OOI-WSR has to be associated with an active representation (agent) of the same OOI. The properties and behavioral models of the software agents corresponding to various OOI types are specified as requirements for the agent-based simulation, using the method put forward in [5,6].

The agent-based simulation runs as an autonomous client-server application, using OOI-WSR for obtaining the initial World State (WS0) and recording the consecutive intermediate World States (WS1 … WSn) back to OOI-WSR. The consecutive WSs reflect the evolution of the World and the emerging states for each agent as a result of interactions between different types of agents (for example, Ambulances←→Patients) and the impact of the environment (e.g. weather conditions, chemical plume, etc.) on these agents. Our agent-based simulation environment supports behavioral models of the following two types:

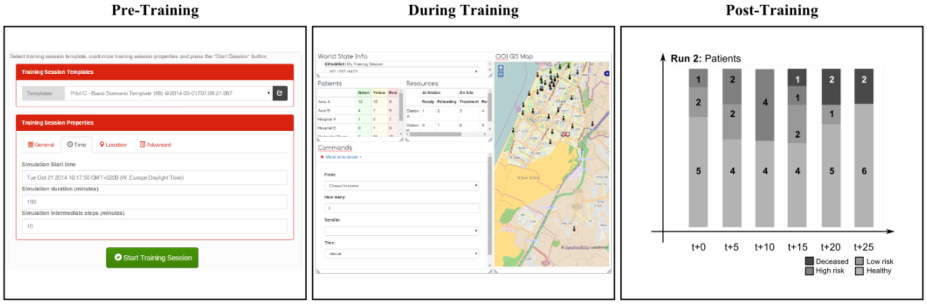
1. Behavioral models for each agent type, such as Ambulance and Patient;
2. Scenarios, specifying "what is played" for different games.

Agent types are designed to be as generic as possible, and the sub-typing (e.g. different ambulances) can be achieved through parameterization at the level of the OOI repository. The agents can also communicate with external models via HTTP or some other fixed protocol. This feature simplifies the calculation/estimation of the agent/environment interactions and also makes agent-based simulation models easily expandable, and allows us to solve rather complicated tasks by means of external services.

However, in many cases different games need game-specific agent types because each agent type must support the scenario(s) foreseen for the game by having all the properties and behavioral models that support realization of the game`s scenarios. In addition, the agent types should also be designed to support the calculation of Key Performance Indicators (KPIs) required by the application owner for decision(s) making for a concrete application. Finally, at the level of scenario, each application of the serious game of resource management requires its specific overall agent-based simulation model developed jointly with the application owner.

3.3 User Interfaces

At the level of graphical interfaces, the application implements three of the "views" foreseen by the CRISMA architecture (**Fig. 4**). The views have been realized as web mashup applications running on a Wirecloud Platform. As a consequence, the whole application is built by "wiring" the simpler GUI widgets which are implemented in JavaScript and HTML5. The application runs in any modern web browser as well as many mobile devices, and the further development and improvements can be easily achieved by replacing or adding new widgets and re-wiring the mashup.



Pre-Training

Select training session template, customize training session properties and press the "Start Session" button

Training Session Templates

Templates: Pilot C - Basic Scenario Template (68) @2014-05-01T07:09:21.087

Training Session Properties

Simulation Start time: Tue Oct 21 2014 10:17:56 GMT+0200 (W. Europe Daylight Time)

Simulation duration (minutes): 150

Simulation intermediate steps (minutes): 10

During Training

Post-Training

Training Setup

- Environment Setup
- Define OOIs
- Define incident location
- Define impact

Resource Management

- Define evacuation zone
- Define safety perimeter
- Dispatch resources
- Monitor resources

Indicators and Analysis

- Evaluate training results
- World state statistics
- Compare decisions

Fig. 4. Main User Interface Views

The *Pre-training View* focuses on the initial creation of a training exercise and allows the user to specify details about the incident from a pre-defined set of templates. This includes the ability to modify each OOI's state and availability. Templates are set up by an expert and already contain a sample configuration of a training exercise but can further be tuned in the pre-training stage. This view is designed to be used by the trainer rather than the trainee.

The *Training View* allows the trainee to play out decisions such as allocation of resources for the training exercise created during the pre-training step. This is achieved by a turn-based approach where the trainee issues any number of orders and then receives the information on a new situation at discreet time intervals. At any point in time, the trainee can issue new orders if deemed necessary, or go back in time and try alternative approaches.

Finally, the *Post-training Analysis View* presents the user with indicators that allow measuring and comparing individual outcomes generated during the training exercise. These comparisons render one or more outcomes as a time series to assess how different decisions played a part in the overall progression. Currently, these statistics cover an aggregated view of the health categorization of patients (healthy, injured, severely injured, and dead), how this status compares to the initial health (improved/deteriorated), as well as the resource utilization over time, but additional indicators can be easily implemented using the Indicator Service.

The fact that results are measurable, and can be correlated with decisions is of major importance for the trainees, as it enables evaluation of the decision processes and not just of the results. In this way, the Analysis View can help the trainees to understand the influence of own decisions on the results.

4 Transferability and the Path Ahead

At a time of writing the article (October 2014), the first version of the training application has been developed and tested by the end-users, and the work on the second application version is underway. Second version of the application is due in March 2015.

The v1 application version already incorporated all the functional elements (**Fig. 2**), OOIs (**Fig. 3**) and GUI elements (**Fig. 4**) which have been introduced in the previous section, but the implementation is still incomplete. In particular, the v2 application will feature: (1) improved patients "life model"; (2) improved interaction between the OOIs (patients, resources) as well as the interaction between OOIs and the environment (e.g. temperature, light conditions, weather, road blocks); and (3) possibility to determine zones with different concentrations and simulate the plume movement during the incident.

Tests of the v1 application with end-users from Magen David Adom (MDA) have confirmed the usability of the application concept and provided a valuable feedback concerning the shortcomings of the design of the graphical interface. For example, the way in which the commands are given to resources in the v1 application differs from the way such commands are given in the command center and the v2 application will be amended accordingly. Nevertheless, all testers agreed that the fully-developed application could enhance the learning experience of the crisis managers' and contribute towards improved preparedness at the MDA. Moreover, the technical development team is confident that the current application design can be easily extended and transferred to new domains and use cases.

Wirecloud offers a solid platform for prototyping and the individual widgets can be easily embedded in other projects at a later time; ICMM simplifies housekeeping and working with indicators and criteria; the Indicator Service simplifies the task of calculating indicators; and the combination of OOI-repository and the Pub/Sub event broker assures that application parts can be developed in parallel without undue overhead.

At the end of the day, the application flexibility remains essentially limited by the functionality of available agent models and by the cost of implementing the missing model features. While no "silver bullet" solution to this issue exists, following features at least mitigate the issue:

- new sub-types of agent models can be introduced in the application by defining a new subtype of OOIs with specific parameters (e.g. max speed, max transport capacity, types of patients which can be treated, etc. for ambulances) within OOI-WSR; and
- as agents can communicate with arbitrary external models, existing models can be easily re-used as is or with small changes where appropriate.

One of the interesting opportunities for future development is in supporting the trainer to choose realistic environmental conditions, e.g. by taking into account past situations and/or forecasting new ones.

Another interesting possibility is extending of the software to support the task of defining and testing of the operational plans. Related work is already underway and the results will be presented in a separate paper later on.

Acknowledgements

The research leading to this paper has been performed within the CRISMA project, which is co-funded under the European Community's Seventh Framework Program FP7/2007 - 2013 (grant agreement no. 284552).

All of the software mentioned in the paper, with the exception of the agents' platform (property of TTU) is licensed under Open Source licenses and can be re-used and extended in other projects.

- Pub/Sub and WireCloud have been developed in the scope of the EC Future Internet Public Private Partnership initiative,
- Remaining software has been developed by CRISMA partners AIT Austrian Institute of Technology, Cismet, and NICE, but may be based on or include third party open source products. Full information on all software used and developed within the project is available at the CRISMA catalogue [9].

References

1. Hawe, G. I., Coates, G., Wilson, D. T., Crouch, R. S.: Agent-based simulation for large-scale emergency response: A survey of usage and implementation. ACM Computing Surveys (CSUR) 45.1: 8 (2012)
2. Dihe, P., Denzer, R., Polese, M., Heikkilä, A.-M., Havlik, D., Sautter, J., Hell, Th., Schlobinski, S.: An architecture for integrated crisis management simulation. In: 20th International Congress on Modelling and Simulation, Adelaide, Australia; Dec. (2013)
3. Dihe, P. et al.: CRISMA framework architecture v2. Project report, http://www.crismaproject.eu/deliverables/CRISMA_D322_public.pdf; (2014)
4. CRISMA consortium, http://www.crismaproject.eu/usecases.htm (2013)
5. Sterling, L. and Taveter, K.: The Art of Agent-Oriented Modeling. MIT Press, Cambridge, MA, and London, England (2009)
6. Miller, T., Lu, B., Sterling, L., Beydoun, G., Taveter, K.: Requirements Elicitation and Specification Using the Agent Paradigm: The Case Study of an Aircraft Turnaround Simulator. IEEE Transactions on Software Engineering, Vol. 40 (No. 10), 1007 – 1024 (2014)
7. Dignum, F.: Agents for games and simulations. Autonomous Agents and Multi-Agent Systems, 24(2), 217-220 (2012)
8. Gentile, M., La Guardia, D., Dal Grande, V., Ottaviano, S., Allegra, M.: An Agent Based Approach to designing Serious Game: the PNPV case study. International Journal of Serious Games 2(1) (2014)
9. Havlik, D., Dihé, P., Frings, S., Steinnocher, K., Aubrecht, C. (eds.). *Catalogue of CRISMA applications, framework building block specifications and software implementations*. CRISMA consortium, https://crisma-cat.ait.ac.at/content/crisma-catalogue-book (2015)