Integrating Discrete Event and Process-Level Simulation for Training in the I-X Framework

Citation for published version:
Wickler, G, Tate, A & Potter, S 2007, Integrating Discrete Event and Process-Level Simulation for Training in the I-X Framework. in *Proceedings of the 4th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2007): Delft, the Netherlands, May 2007.* VUB Press. <http://www.iscramlive.org/dmdocuments/ISCRAM2007/Proceedings/Pages_355_359_41ASCM_08_A_Integrating.pdf>

Published In:
Proceedings of the 4th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2007)

Wickler, G., Tate, A., & Potter, S. (2007). Integrating Discrete Event and Process-Level Simulation for Training in the I-X Framework. In Proceedings of the 4th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2007). VUB Press.

Integrating Discrete Event and Process-Level Simulation for Flexible Training in the I-X Framework

Gerhard Wickler
AIAI, University of Edinburgh
g.wickler@ed.ac.uk

Austin Tate
AIAI, University of Edinburgh
a.tate@ed.ac.uk

Stephen Potter
AIAI, University of Edinburgh
s.potter@ed.ac.uk

ABSTRACT

The aim of this paper is to describe I-Sim, a simulation tool that is a fully integrated part of the underlying agent framework, I-X. I-Sim controls a discrete event simulator, based on the same activity model that is shared between all I-X components, and multiple process-level simulators that model the continuous change caused by actions that are considered as primitives by the rest of the system. The primary purpose of this tool is to support instructors during exercises that are used for training in emergency response. The main advantage the I-Sim tool gives the instructors is flexibility, allowing them to orchestrate and modify existing training scenarios on the fly, adapting them to trainees' needs as required.

Keywords

HTN planning; Discrete Event Simulation, Emergency Response Training, Personnel Recovery

INTRODUCTION

If we are lucky, we never have to use any of the emergency response systems we are building. However, reality is not that kind and we need these systems occasionally. Some systems will be in relatively regular use, e.g. by a fire brigade or a police force, whereas others will hardly ever be in real use due to the nature of the emergency being infrequent, e.g. a major earthquake. The same applies to the procedures that need to be implemented in such emergencies and that information technology/command & control systems are meant to support. As a result, there is a training issue, at least with the less frequently used systems and procedures. Without regular practice, people tend to forget how to use a system or what to do in a given situation, and there is no time during an emergency to go through manuals describing pre-defined response plans or how to use some software.

This leads to a need for repeated training so that emergency responders are at all times familiar with the appropriate procedures and plans as well as with the software systems they need to use. Training is often given in the form of exercises or games in which a realistic emergency is simulated and responders have to deal with it as if it were real (Turoff *et al.*, 2006). This requires a scenario to be designed and executed by the instructors, who should be experts in the field. Experts, however, are often hard to find and do not have much time. Reusability of scenarios is limited and trainees should be presented with different scenarios every time they are being trained, thus requiring even more expert time. Finally, even experts have difficulties predicting how certain types of disasters develop over time, e.g. a large fire or an oil spill. Mathematical computer models can be much better for simulating this type of continuous process, but they lack the creativity of human experts when it comes to scenario design.

The *I-X* framework (Tate, 2000; Tate, 2003) described briefly in this paper includes a tool that can support instructors in addressing some of the issues raised above. It provides a fully integrated discrete event simulator using a shared representation that can be easily scripted at the knowledge level for scenario development and modified during the execution of such a scenario in a training session. It also provides flexibility in the sense that trainers can diverge from the script during a training exercise, thus allowing for an element of creativity and reaction to the trainees' actions. Furthermore, it allows for the seamless integration of other types of simulators, e.g. those that use mathematical modelling techniques to realistically model the processes and events that occur during an emergency. This I-X tool is called *I-Sim* and was developed for various purposes:

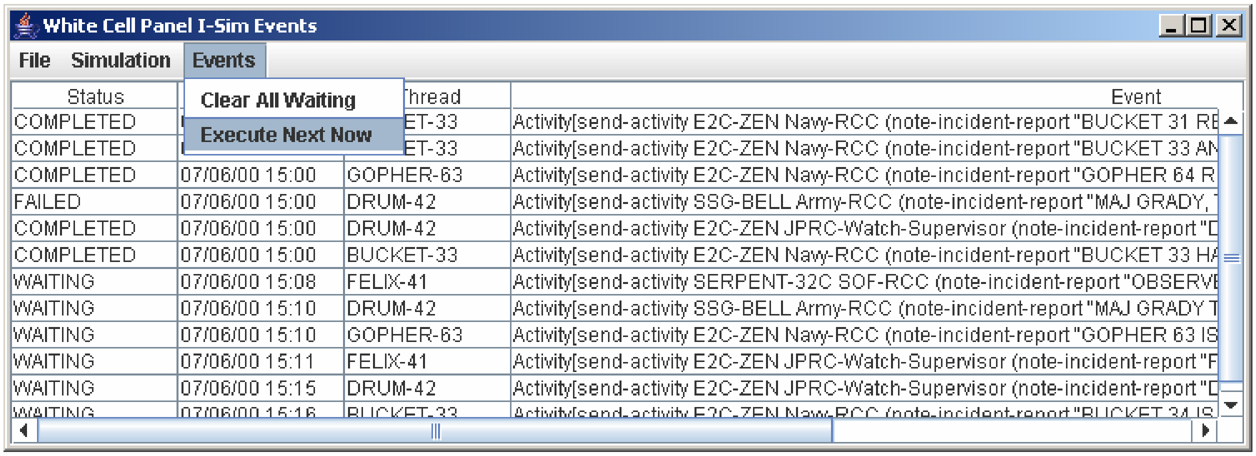
- for education and training in emergency response: the simulator can be used to execute scenarios and emulate a realistic unfolding of events, but instructors can always change the course of events to optimize the learning experience;
- for testing and evaluation of emergency plans in simulated scenarios: the simulator can be used to emulate some events and standard response plans can be compared in these events, one after the other, thus highlighting their pros and cons or showing up gaps; and
- for integrating a mathematical model into the real response system: during an actual incident the simulator can be used to perform a super-real time simulation, providing responders with a way to look into the probable future.

I-Sim is being used to simulate the real world for execution monitoring and feedback purposes. This contrasts with another potential use of simulation to provide plan analysis at plan time to feedback information to allow for rating and refinement of plan options (e.g. as in Dyke *et al.*, 2000).

THE I-X FRAMEWORK

There are a number of tools available that help people organize their work. One of these is provided with virtually every organizer, be it electronic or paper-based: the "to-do" list. This is because people are not good at remembering long lists of potentially unrelated tasks. Writing down these tasks and ticking them off when they have been done is a simple means of ensuring that everything that needs to be done does get done, or at least, that a quick overview of unaccomplished tasks is available. In responding to an emergency this is vital, and the larger the emergency, the more tasks need to be managed.

I-X is a framework that can be used to create an application in which multiple agents, be they human or software, adopt a task-centric view of a situation, and which supports the necessary coordination of their activities to respond to that situation. The I-X Process Panel (Tate *et al.*, 2002; Potter *et al.*, 2006) provides the functionality of a to-do list and thus could be a useful tool when it comes to organizing the response to an emergency. The idea of using a to-do list as a basis for a distributed task manager is not new (Kreifelts *et al.*, 1993). However, I-X goes well beyond this metaphor and provides a number of useful extensions that facilitate the finding and adaptation of a complete and efficient course of action. I-X is being developed at the University of Edinburgh and is based on an HTN planning architecture that represents the experience gained in more than 3 decades of AI planning research.



Status		hread	Event
COMPLETED		ET-33	Activity[send-activity E2C-ZEN Navy-RCC (note-incident-report "BUCKET 31 RE
COMPLETED		ET-33	Activity[send-activity E2C-ZEN Navy-RCC (note-incident-report "BUCKET 33 AN
COMPLETED	07/06/00 15:00	GOPHER-63	Activity[send-activity E2C-ZEN Navy-RCC (note-incident-report "GOPHER 64 R
FAILED	07/06/00 15:00	DRUM-42	Activity[send-activity SSG-BELL Army-RCC (note-incident-report "MAJ GRADY,
COMPLETED	07/06/00 15:00	DRUM-42	Activity[send-activity E2C-ZEN JPRC-Watch-Supervisor (note-incident-report "D
COMPLETED	07/06/00 15:00	BUCKET-33	Activity[send-activity E2C-ZEN Navy-RCC (note-incident-report "BUCKET 33 HA
WAITING	07/06/00 15:08	FELIX-41	Activity[send-activity SERPENT-32C SOF-RCC (note-incident-report "OBSERVE
WAITING	07/06/00 15:10	DRUM-42	Activity[send-activity SSG-BELL Army-RCC (note-incident-report "MAJ GRADY T
WAITING	07/06/00 15:10	GOPHER-63	Activity[send-activity E2C-ZEN Navy-RCC (note-incident-report "GOPHER 63 IS
WAITING	07/06/00 15:11	FELIX-41	Activity[send-activity E2C-ZEN JPRC-Watch-Supervisor (note-incident-report "F
WAITING	07/06/00 15:15	DRUM-42	Activity[send-activity E2C-ZEN JPRC-Watch-Supervisor (note-incident-report "D
WAITING	07/06/00 15:16	BUCKET-33	Activity[send-activity E2C-ZEN Navy-RCC (note-incident-report "BUCKET 34 IS

Figure 1: I-Sim window with events from multiple threads during simulation.

THE I-SIM TOOL

The I-Sim tool integrates the functionality of a discrete event simulator and a number of process-level simulators into the I-X framework. Usually there will be only one I-X Process Panel that has access to this tool, e.g. the panel representing the instructors in a training scenario. In such a set-up the I-Sim tool gives the instructors the necessary control over the simulation, while other panels that do not have access to the tool may only see the result of the simulation. As a result control over the development of the scenario is centralized, whereas I-X panels supporting responders are expected to be distributed. An example I-Sim tool is shown in figure 1, displaying events taken from the Co-OPR application described below.

The tight integration of the I-Sim tool with the I-X framework is achieved through use of a shared model of activity termed <I-N-C-A> (Tate, 2003) that is maintained in the panel to which the tool belongs. Thus, the simulator can extract all required information to start the simulation directly and without necessitating any user interaction. Furthermore, updated state information can be written back to this shared model, thus allowing all the attached viewers to display this information in a consistent way, i.e. shared state information is visible in all other tools. Finally, state can also be shared with other panels allowing students selective access to simulation results. The impact of these design choices is that shared models do not give rise to undesired inconsistencies that could disrupt a training exercise and increase maintenance overheads.

For a training session the scenarios have to be scripted in advance, with each incident to be addressed corresponding to a sequence of events. The I-Sim tool allows the user to load one or more such incident scripts at a time, thus allowing the combination of different incidents. Events are associated with the incident they were loaded from and the tool displays this as a ‘thread’ running through the scenario, information that is not available to the trainees. Note that this means that multiple incidents and their simulation can take place in parallel. Each event is ‘timed’ and the time represents the start time of the event relative to the start of the incident. The ‘actual’ start time of the simulation has to be defined when an incident is loaded. Additional incidents that are added into the scenario after the start of the simulation will begin at the time of loading. This gives instructors the flexibility to extend the scenario quickly and according to desired training outcomes.

Another aspect of the simulation that can be controlled through the I-Sim tool is simulated time. At the beginning of the simulation the time acceleration factor can be set to determine how fast the simulation will progress. This can be any positive number, including numbers less than one (meaning simulation time passes slower than real time). The time acceleration factor currently can only be set at the beginning of the simulation. However, the simulation can be paused and resumed at any point. Jumping forward and backwards is not possible as this has technical implications for the process-level simulators used, i.e. they too need to support such jumping, and this may not always be feasible. It is possible to skip forward to the next event, but this means that this event will be injected into the scenario now rather than at its scheduled time. The intended result of these time-management facilities is a reduced cognitive load on the instructors.

The current status of events is always shown in the I-Sim tool window, thus giving instructors a quick overview.

The aim of the above controls is to give the instructors flexibility. This is achieved by giving instructors the option to create a scenario on the fly from pre-defined components and add more incidents at a later point if required. Furthermore, instructors have some limited control over simulated time by setting the time acceleration factor and pausing and resuming the simulation to adapt to the trainee’s progress.

Discrete Event Simulation

The discrete event simulator provided by the I-Sim tool is tightly integrated with the I-X framework. The simulation of actions is based on the same representation in the domain model used by all the components and tools in the framework and can be edited using the integrated domain editor. This domain model is based on a hierarchical task network (HTN) planning (Ghallab *et al.*, 2004) representation and consists mainly of refinements and actions. Actions are primitives in the sense that the planner has no further knowledge about the internal structure of these activities; it only looks at actions at the envelope level where they are described in terms of preconditions and effects. The choice of an HTN model for activity representation allows for a description of unfolding incidents at various levels of abstraction, e.g. strategic, tactical and operational event and action sequences. HTN planning also has been used successfully in a number of realistic domains.

When I-Sim simulates an action, its default behaviour is to use only the preconditions and effects of the action. First, the preconditions are evaluated against the current state of the world as represented in the agent’s current <I-N-C-A> model. For the simulation to be able to start, the preconditions have to be uniquely satisfiable, that is, there must be exactly one way to bind all the remaining variables in the preconditions. If there is no way to bind the variables consistently, the action cannot be applied. If there is more than one way there is still choice that the simulator cannot make. If an action can be simulated, the effects of the action are asserted just before the simulation is terminated. This ensures that the simulation is consistent with the domain model to which instructors and students have access during the exercise.

Actions may be instantaneous, e.g. sending a message to another panel notifying it that an event has been observed, or they may take time as specified in the action description, e.g. flying a rescue helicopter from one location to another. If a time is specified the default behaviour of the simulator is to wait for the specified amount of time before the effects are asserted. This does not result in any change of world state during the execution however, which is simply due to the fact that there is no further information available to the discrete event simulator.

Process-level Simulation and Time

Discrete events are often sufficient from a controller's perspective, but finer grained models are required to simulate the development of a situation, e.g. a fire, an oil spill, or the spreading of a virus. As opposed to discrete event simulations, process-level simulators, often based on mathematical models, emulate a world that appears to be changing continuously.

In I-Sim, process-level simulators can be attached to actions in the domain model. When asked to simulate an action, I-Sim first verifies the preconditions as described above. Then, instead of simply waiting, it executes a process-level simulation which may (periodically) assert facts into the simulator's state to simulate the continuing change of the world. Finally, I-Sim asserts the effects of the action as for the discrete event simulation. As a result the simulation progresses naturally and continuously. Furthermore, the use of mathematical models means that events and resulting world states may be more realistic and contain more detail than a human could generate. However, I-Sim has no control over the quality of these process-level simulations.

Note that I-Sim allows for multiple simulations to be run in parallel. Interference/interaction between different simulators is coordinated via the shared <I-N-C-A> model.

The Co-OPR Application

When the I-X framework is instantiated with a domain-specific model, we refer to it as an I-X application. In such an application the Process Panel can function as an interface for the (human) user or as a semi-autonomous agent acting on behalf of the user. Such an application has been developed during the Co-OPR project for the task of personnel recovery training.

Personnel Recovery (PR) is the sum of military, diplomatic and civil efforts to effect the recovery and reintegration of isolated personnel. During any military operation Joint Force Commanders and Staff are responsible for and must be prepared to accomplish any PR tasks occurring within a specified operational area or else determine and accept the risk of not doing so (Joint Publication 3-50, 2005). In order to be prepared, the USJFCOM/JPRA Personnel Recovery Education and Training Center (PRETC) in Fredericksburg, VA, trains US military personnel in the execution of PR tasks. This training consists of classroom sessions, in which the necessary 'textbook' knowledge is taught, and of Command Post Exercises (CPX), in which the students have to perform PR tasks during a simulated fictitious military operation. Events related to incidents are currently injected manually by the trainers.

Co-OPR uses I-Sim to support the instructors in a CPX scenario. The scenario simulated in a CPX is defined by the planned events in the Air Tasking Order (ATO), which is known to the students, and by the Master Scenario Events List (MSEL) defining the incidents that will occur. Events in the MSEL are mostly observations related to incidents that the students have to keep track of and act upon in an appropriate way. With I-Sim instructors cannot forget or wrongly delay the injection of events (simulated time is always known to all participants due to synchronized I-Sim clock). Events from the MSELs can also be sent as messages to panels automatically, or reminders can be given to instructors when telephone interjections are required.

At present, the MSELs used in the CPX are written as Word documents and cannot be changed during the (manual) simulation. With I-Sim the instructors can change the scenario during an exercise in various ways, thus giving them the flexibility to add incidents to the scenario if the students are doing well, and so keep the exercise challenging. Furthermore, the process-level simulations allow the addition of more detail to the scenario, e.g. a helicopter flying to rescue a person has an automatically maintained position that can be monitored, and taken into account if new incidents occur that require a change of plan.

CONCLUSIONS

The I-Sim tool is currently used in a number of projects including Co-OPR described here. A number of experiments have been conducted to evaluate the system (Wickler *et al.*, 2007). A final experiment is planned for May 2007 focusing on the I-Sim tool that is to be used in a real CPX held at the Personnel Recovery Education and Training Center in Fredericksburg, VA. The aim of this application is to give instructors more flexibility during a CPX while decreasing the mental load. The idea is to run the I-Sim tool in parallel to the normal running of a CPX, but only with a single trainer (as opposed to 4 for the normal CPX). The reduced cognitive load should make this possible. Furthermore, the trainer should have the time to extend the original scenario at a sensible point in the exercise. More generally, the aim of I-Sim is to save expert time by simplifying the implementation and execution of new scenarios for training exercises. This can be achieved by recombining incidents and different types of simulators. The tight integration of the I-Sim tool means that any changes made by the instructors immediately propagate throughout the application, i.e. to all the panels–and hence to all the human agents–involved, and in an appropriate way.

ACKNOWLEDGMENTS

The I-X project is sponsored by the Defense Advanced Research Projects Agency (DARPA) under agreement number F30602-03-2-0014. The University of Edinburgh and research sponsors are authorized to reproduce and distribute reprints and on-line copies for their purposes notwithstanding any copyright annotation hereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of other parties.

REFERENCES

1. Dyke, D.S., Salt, M.G., Jarvis, P. and Desimone, R. (2000) Experimental Results from Integrating Planning Systems and Simulation Models, 2000 Command and Control Research and Technology Symposium, in Proceedings of the US Naval Postgraduate School, Monterey, California, USA, June 26-28th, 2000.
2. Ghallab M., Nau D., and Traverso P. (2004) *Automated Planning – Theory and Practice*, Morgan Kaufmann.
3. Joint Publication 3-50 (2005) Joint Doctrine for Personnel Recovery, 2nd Draft, March 2005.
4. Kreifelts Th., Hinrichs E., and Woetzel G. (1993) Sharing To-Do Lists with a Distributed Task Manager. In: de Michelis G. and Simone C. (eds.) *Proceedings of the 3rd European Conference on Computer Supported Cooperative Work*, pp 31-46, Milano, 13-17 September 1993, Kluwer, Dordrecht.
5. Potter, S., Tate, A. and Wickler, G. (2006) Using I-X Process Panels as Intelligent To-Do Lists for Agent Coordination in Emergency Response, Proceedings of the Information Systems for Crisis Response and Management 2006 (ISCRAM2006), Newark, New Jersey, USA, May, 2006.
6. Tate, A. (2000) Intelligible AI Planning. In: *Research and Development in Intelligent Systems XVII, Proceedings of ES2000*, The Twentieth British Computer Society Special Group on Expert Systems International Conference on Knowledge Based Systems and Applied Artificial Intelligence, pp. 3-16, Cambridge, UK, December 2000, Springer.
7. Tate A. (2003) <I-N-C-A>: An Ontology for Mixed-initiative Synthesis Tasks. *Proceedings of the Workshop on Mixed-Initiative Intelligent Systems (MIIS)* at the International Joint Conference on Artificial Intelligence (IJCAI-03), Acapulco, Mexico, August 2003, pp 125-130.
8. Tate, A., Dalton, J., and J. Stader, J. (2002) I-P2- Intelligent Process Panels to Support Coalition Operations. In *Proceedings of the Second International Conference on Knowledge Systems for Coalition Operations* (KSCO-2002). Toulouse, France, April 2002.
9. Turoff, M., Chumer, M. and Hiltz, S.R. (2006) Emergency Planning as a Continuous Game, Proceedings of the Information Systems for Crisis Response and Management 2006 (ISCRAM2006), Newark, NJ, USA, 2006.
10. Wickler, G., Tate, A. and Potter, S. (2006) Using the <I-N-C-A> Constraint Model as a Shared Representation of Intentions for Emergency Response, Proceedings of the First International Workshop on Agent Technology for Disaster Management (ATDM), at the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006), Future University, Hakodate, Japan, May 8-12, 2006.
11. Wickler, G., Potter, S., Tate, A. (2007) Using I-X Process Panels as Intelligent To-Do Lists for Agent Coordination in Emergency Response, International Journal of Intelligent Control and Systems (IJICS), Special Issue on Emergency Management Systems.