Simulation Case study

A simple distributed architecture for emergency response exercises

- A simple distributed simulation for support of emergency response exercises.
- Immersive Synthetic Environment for Exercises (ISSS).
- Inspired by web-based role playing games (RPGs).
- Developed by the Institute for Security Technology Studies (ISTS) at Dartmouth College.
Case study

• This simulation was developed as a prototype to support a mass-casualty emergency response in 2005.

• Results of the exercise showed that the intended value of distributed simulation was achieved using simple tools with a rapid development time.

• Inspired by web based role playing games, ISEE uses:
  ✓ PHP (for user interface)
  ✓ MySQL (as a middleware)
  ✓ Python (as a simulation engine)

Background

• We live in an age of acute awareness of the potential for catastrophic events.

• The most effective way to prepare for a response and recovery for such events is through emergency response exercises.

• Traditionally these exercises have been done as full-scale exercises or else through table top exercises.
Case study

- **Full-scale exercise**
  - Requiring the costly deployment of vehicles, personnel and equipment.

- **Table top exercise**
  - Players around the table vocalize their responses to the scenario, with the scenario changing in response to their actions and random factors which may be determined by rolling dice or drawing cards.
  - Much less expensive than full-scale exercise.

ISEE Architecture

- ISEE provides a framework for a distributed emergency response simulation.

- Emergency response was compared to role playing games.

- RPGs involve fantasy-based characters such as wizards, dwarfs and elves collectively fighting against monsters with some combination of weapons and spells.
Case study

• Emergency response simulation involves different kinds of characters collectively applying resources to solve problems.
  ✓ Ambulances, fire engines, etc...
  ✓ Treatment of casualties, fire suppression, etc...

<table>
<thead>
<tr>
<th>RPG</th>
<th>Emergency Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Players</td>
<td>Fire, Police, Incident command</td>
</tr>
<tr>
<td>Inventory</td>
<td>Fire engines, Ambulances, Police cruisers</td>
</tr>
<tr>
<td>Challenges</td>
<td>Fire suppression, casualty management, public safety</td>
</tr>
<tr>
<td>Communication</td>
<td>Radios, mobile phones</td>
</tr>
</tbody>
</table>

The simplicity of web-based RPGs inspired a simple architecture for ISEE which retains the fundamental components of distributed simulation while satisfying the cost and simplicity needs of the emergency response community.

The result is a distributed simulation which allows incident commanders to rehearse emergency response procedures in real-time, with very little computing resources.
The three-tiered structure of ISEE

- Html-based user interfaces
  - Generated using PHP
- A MySQL middle layer
  - Keeps track of persistent simulation object state
  - Saves all events and communications for after action review.
- Agents and objects written in Python
  - Agents and simulations can connect to the database either locally or remotely
  - Object state can be modified by users through the web interface and by the Python agents.
Case study

User interface

• The user interface of ISEE is based on three principles of incident management:
  ✓ Situational awareness
  ✓ Communication
  ✓ Resource management

• Users log on as one of the several roles (e.g. police, incident command, etc...)

• Each role has a page dedicated for it.

Situational awareness comes primarily from the an event window which notifies users of significant events that occur during the simulation.

Additional situational awareness is derived from maps and images of the incident scene as well as weather conditional and an elapsed time clock.

Radio communication is done through a text massage panel, where users types messages and send them over to several channels (police, fire, broadcast).

✓ Every message is logged in the database for the after action review.

Matthew Xuereb
Department of Computer Science
Case study

- Resources are shown and managed through a pop-up window.

- Resources includes:
  - People
  - Equipment
  - Vehicles

- Such resources can be tasked to address specific problems such as:
  - Fire suppression
  - Traffic control
  - Casualty treatment

Case study

- Each user (role player) is equipped with a set of functions to help them complete their responsibilities per the disaster response plan.

- Users are also provided with visual indicators that allows them to see how they are progressing against their specific challenges (i.e. Fire, traffic, ...).

- Some objects (resources) are owed by different users during the course of the simulation.
Case study

• Example:
  ✓ An ambulance initially belongs to dispatch who sends it to the scene.
  ✓ When it arrives on the scene, ownership is transferred to the incident command.
  ✓ Incident command then has the option of assigning it to various branches (such as police, fire, medical,...).
  ✓ Then the branch commander has the control over that resource.

• Users only have visibility and control over resources that they own.

OSEE in action

• A prototype ISEE system was used in May 2005 to simulate a mass casualty exercise in Lebanon.

• The scenario involved an airport crash which was mandated by the FAA to maintain airport certification.

• The simulation involved an aircraft fuel fire and 20 casualties with injuries ranging from superficial to fatal.
Case study

• Players in the simulation included:
  ✓ Airport operations
  ✓ Indecent command
  ✓ Police
  ✓ Fire
  ✓ Emergence Medical Operations
  ✓ Hospital

• Resources for the exercise were based on actual vehicle inventories and capabilities.

Case study

• The simulation time was just under one hour from the report of the plane crash to dispatch until the transport of all living casualties from the scene.

• 21 airport vehicles and 15 aid vehicles form neighbouring communities were called to respond.

• 118 simulation events and 159 communication messages were logged.

• Users said that the level of interaction and realism exceeded that of a traditional tabletop exercise.
Case study

• Negative feedback from user focused about the communication panel – chat application cues very confusing.
  ✓ An improvement would be that of using audio cues instead of visual

• Future research will be conducted into methods for capturing voice communications so that users will be able to communicate naturally.

Conclusions

• ISEE demonstrates that a simple distributed simulation based on open tools can be effective method for improving tabletop exercises.

• Its lack of dependence on custom or proprietary tools and its ability to run on a web browser make it ideal for emergency response exercises.
  ✓ Enables users at different level of response to participate in distributed exercises without travelling to a common exercise venue.
Case study

• The use of MySQL as the middle ware is not the solution for all distributed problems
  ✓ It does not support any time management
  ✓ Not able to support vary large simulations

• However for small simulations (with less than 1000 entities) it can be used effectively.
  ✓ Low cost

Bibliography

Paper