Ensayo: A Distributed Web-based Virtual Emergency Operations Center Prototype

Cynthia Nikolai
University of Notre Dame
Computer Science & Engineering
College of Engineering
Notre Dame, IN, 46556 USA
cnikolai@nd.edu

Greg Madey
University of Notre Dame
Computer Science & Engineering
College of Engineering
Notre Dame, IN, 46556 USA
gmadey@nd.edu

Irma Becerra-Fernandez
Florida International University
Decision Sciences and Information Systems
College of Business Administration
Miami, FL, 33176, USA
becferi@fiu.edu

Keywords: crisis management, emergency operations, training simulator, expert systems, rules-based systems

Abstract

"It has been argued by researchers that under a crisis situation, humans tend to make decisions more "naturally" or "intuitively" rather than wholly "rational." It is believed that experience and extensive practice dealing with uncertainly and time stress within a particular domain is the only way to improve crisis decision-making skills."[4] In the current crisis management arena, most training is conducted via live or face-to-face exercises [3]. This means that training requires participation from many individuals, consumes a great deal of money, and cannot be conducted often. In this paper, we describe a socio-technical emulator and training facility for upper level emergency managers. This tool is important because it enables emergency managers to train under crisis conditions in a virtual arena. It also serves as a research tool for cognitive scientists to study the decision making process under emergency conditions.

1. INTRODUCTION

Hurricane Katrina was one of the most expensive and devastating natural disasters in American history [5]. Over half a million people were affected by the hurricane, and the US energy infrastructure was severely damaged [5]. Hurricane Katrina and other natural disasters clearly show the need for improvements in crisis management, especially in training and collaboration among federal, state, and local governments [3, 5]. There are a number of limitations to current non-computer-based training solutions. First, crises, by definition, are rare events, and therefore they do not lend themselves to enable extensive training. Moreover, in the middle of a crisis, few organizations have the time or resources to train new personnel; Their foremost focus is on stabilizing the crisis, not on training individuals [1]. Another limitation of face-to-face solutions is that there are few experts even available, and each expert is inherently constrained by limited time, experience, and perspective [1]. In addition, there is the difficulty of training teams, training selective components of the crisis hierarchy, and training upper level managers [1,2]. In fact, while there are multiple computer-based solutions available for first responders, current research identifies a general lack of computer-based training that targets upper level crisis managers [3]. Moreover, the training that does take place can be ineffective because most instructors use subjective measures and usually end up emphasizing outcomes over decision management processes. Finally, in face-to-face and instructor-centric solutions, there are usually inherent time delays in the feedback as experts analyze the student's progress, compare the student's actions and outcomes to the expected actions and outcomes, and tailor the feedback to the individual [1].

2. OVERVIEW

2.1. Computer-Based Solutions to Emergency Management

Computer-based solutions, on the other hand, can adequately address these limitations. Computer-based training allows emergency managers to train new personnel without being in the middle of a disaster. Moreover, they also allow personnel to train more frequently than they otherwise would be able to in live and face-to-face exercises. In addition, they enable distributed access to data, resources, communication, and training. Computers offer another advantage: in programming the training system for evaluation, we are effectively gathering a wide base of experts, which allows us to reap the benefit of a collective knowledge base of expertise, rather than that of one or two individual instructors [1]. Computers also enable us to train teams and to target and train selective portions of the emergency management personnel and hierarchy.

Finally, whereas feedback has delays in non-computer solutions, feedback can be immediate in a computer-based system.

2.2. Project Ensayo: a Distributed Web-based Virtual Emergency Operations Center (vEOC)

In this project, we incorporate computer-based solutions in emergency management by creating a distributed web-based simulator of an Emergency Operations Center (EOC). An EOC is a secure location in which upper level emergency managers come together to prepare for, manage, and coordinate recovery activities in response to an emergency situation (eg hurricane, earthquake, tsunami). There are several key features of this software. First, it targets upper level emergency managers, hereafter referred to as emergency managers. Second, it allows emergency managers to access databases, to coordinate emergency response, and to train in a virtual area. In addition, since our system is web-based and distributed, it allows mangers to train in a remote manner as well. That is, by its nature, the vEOC enables emergency managers to train from a physical location such as an EOC. However, if some individuals are not able physically to come to the EOC or training location, then they can still train in the vEOC with the rest of the group via remote login and participation. Third, while the vEOC enables EOC personnel to train when all of the personnel are available, one of the key features that makes our virtual environment unique is that it enables emergency managers to train not only locally and remotely, but also asynchronously with respect to other trainees. That is, we augment the environment with artificially intelligent agents to facilitate training whenever all of the trainees are not available. In this way, training can continue when one or more individuals cannot be present for the exercise.

2.3. Expert Validation

In order to validate the system and obtain an expert subject matter knowledge base, we are working with one of the foremost emergency operations centers in the country – the Miami-Dade County EOC in Miami-Dade County, Florida.

3. RESEARCH METHODOLOGY

The basic research methodology has been as follows. First, we began with a literature review of emergency management, simulation, training, and decision making. Next, we coordinated with Miami-Dade County for real world procedures and operations. Third, we began observing the EOC in live exercises. Finally, we were ready to begin building the prototype. Along the way, we have been soliciting feedback directly from subject matter experts at Miami-Dade and updating the vEOC accordingly. We also have had the opportunity to witness a live EOC activation, from which we were able to incorporate several additional expert feedback suggestions.

4. DESIGN METHODOLOGY

There are four main design methodologies that we employ in creating this prototype: spiral development, model-view-controller, mental models, and user-centered application design.

4.1. Spiral Development

The basic software design methodology is the spiral development model [6]. This is the software engineering model in which we design and develop the prototype in iterations in which we incorporate feedback from the previous phase into the current phase.

In our development, we have split the design into 3 primary iterations(see figure 1). The blue circles indicate the amount of effort that is required in each iteration during each phase of the software engineering (ie Design, Build, and Evaluate). The larger the circle implies that more effort is required. The orange chevrons indicate the start of new iterations. The yellow 4-pointed stars indicate completed validation points, and the purple 5-pointed starts indicate planned validation points. In the first iteration, we designed and deployed a single client to train in a virtual environment. Then we solicited feedback from EOC personnel in our design [12,13] and implementation [14,15]. In iteration #2, we make the vEOC multi-client and we supply server push (see section 7.1). After iteration #2, we will solicit feedback to validate the design again.

In iteration #3, we will begin implementing artificial agents to supplant humans that are not available to participate in the exercise.

4.2. Model-View-Controller

In terms of our actual design, we employ the model-view-controller (MVC) framework. The MVC framework is a design methodology in which the way we display the data to the user (the view) is abstracted from the data itself (the model) as well as from the way we manipulate the data (the controller).[7]

4.3. Mental Models

A third design methodology we employ is mental models. Mental models are a tool to aid in user-centered design and a way to ensure that all functionality in our prototype maps to a genuine user need.[8] Essentially, we query the users to ascertain their functional needs (mental models) and then we try to map all functionality in our system to those mental models.

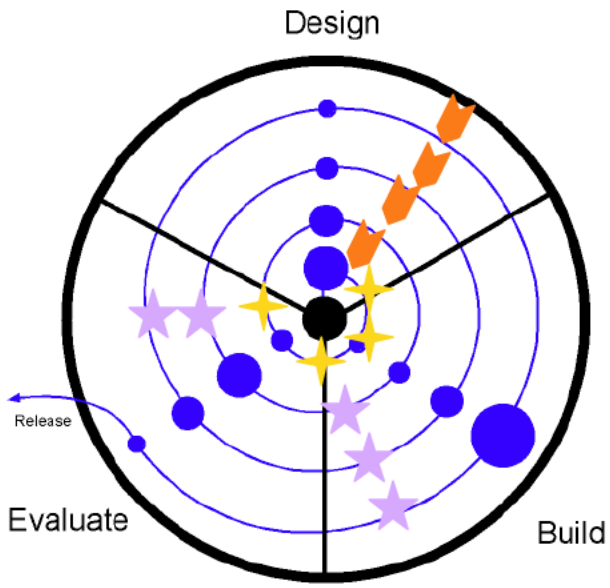

Figure 1: vEOC Spiral Design Methodology[1]

4.4. User-Centered Application Design

Finally, we also use content, functionality, aesthetics, and usability design methodologies in our application design. As summarized by Prof Christopher Clark based on his research, these four methodologies are key ingredients of user interface design. [9] Basically, content includes the features that are on the user interface, aesthetics indicates how pleasing the user interface is to the eye, functionality includes what the user interface is capable of doing, and usability is the "user-friendliness" of the interface.

5. USER VIEWS

There are 6 different user views in the vEOC, that is, there are 6 main roles that a user may exercise: the trainee, the observer, the scenario manager, the staff member, the administrator, and the researcher.

5.1. Trainee

The trainee prepares for emergency situations and practices decision making by interacting with the vEOC. The trainee has the ability to modify all status in the vEOC.

5.2. Observer

The observer prepares for emergency situations by watching the trainees and other personnel in the vEOC; He/she does not have the ability to modify status in the vEOC other than to read and search through archives.

5.3. Scenario Manager

The scenario manager creates scripts to train emergency personnel. The scenario manager also moderates the exercise/training sessions. The scenario manager is essentially equivalent to the controler in the functional exercise, with the exception that injects are automatically presented to the trainee. The scenario manager has the greater ability/responsibility to begin, pause, and terminate the exercise. The scenario manager can also speed up or slow down the exercise as well.

5.4. Administrator

The administrator maintains the vEOC software. The administrator also sets up and moderates user profiles.

5.5. Staff Member

Staff members are upper level EOC staff. For example, they may be EOC planning section personnel. Staff members can print reports and analyze performance and decision making of the EOC personnel.

5.6. Researcher

Researchers are individuals interested in studying various aspects of decision making and emergency response. They typically are not EOC personnel.

6. MENTAL MODELS

The corresponding mental models of each of the user roles are illustrated in figures 2-7. This functionality is based on literature review and observation of the Miami-Dade EOC.

6.1. Trainee

The trainee needs to be able to see the status of the current exercise; view, update, and react to various injects; continue the exercise despite other computer/trainee actions and failures; create, save, import, export, and manage reports. This includes printing regular reports as well as templates for ad-hoc reports. The trainees also need to integrate additional resources with the vEOC. In the Miami-Dade EOC, additional resources include e-Team, Hurrevac, SALT, and SLOSH. Finally, there needs to be a place for both primary and secondary participants in the exercise. That is, sometimes a station in Miami-Dade will have both a primary and a secondary representative heading the station. [12,14]

---

[1] Adapted from http://faculty.washington.edu/farkas/TC407/SpiralModel.gif

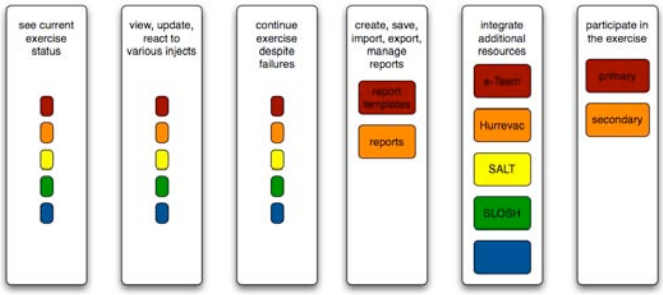

Figure 2: Mental Model Trainee

6.2. Observer

The observer has the same functionality as the trainee with one main exception. He/she does not have the ability to modify status in the vEOC other than to read status and search through archives**.**

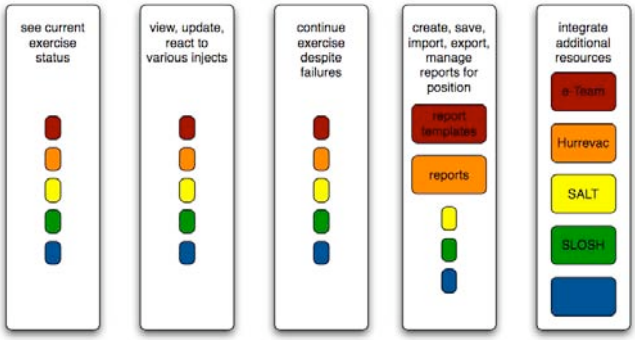

Figure 3: Mental Model Observer

6.3. Scenario Manager

The scenario manager imports, exports, and manages scripts that train emergency personnel. This includes creating, editing, and saving scripts. The scenario manager also moderates the exercise/training sessions by interacting with the vEOC console. This includes functionality such as start, stop, pause, resume, next block, fast time, and normal time. The scenario manager also needs the ability to continue the exercise when all of the trainees are not available.

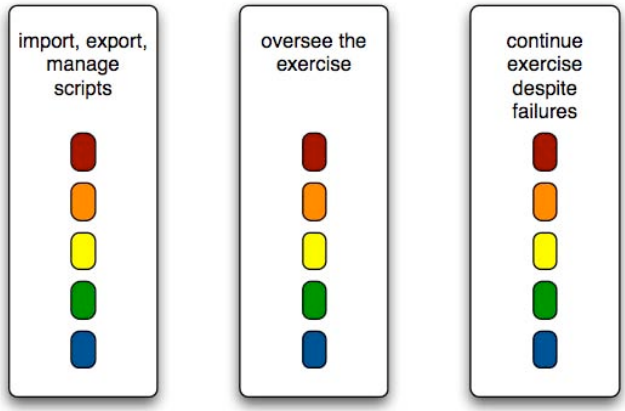

Figure 4: Mental Model Scenario Manager

6.4. Administrator

The administrator installs, configures, manages, and maintains the vEOC software. The administrator also sets up and moderates user profiles and to update user profiles dynamically (when trainees log in/log out of the vEOC).

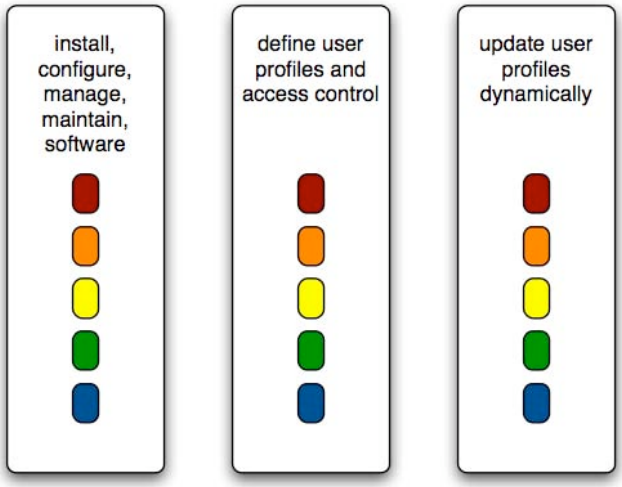

Figure 5: Mental Model Administrator

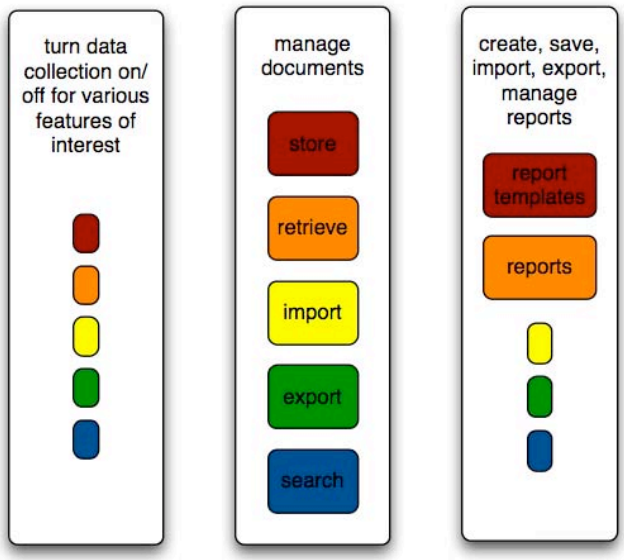

Figure 6: Mental Model Staff Member

6.5. Staff Members

Staff members can turn data collection statistics on and off for various features of interest. Staff members can also manage documents. This includes storing, importing, exporting, retrieving, and searching. Finally staff members can create, save, import, export, and manage reports as well. Again, this includes regular reports as well as ad-hoc reports.

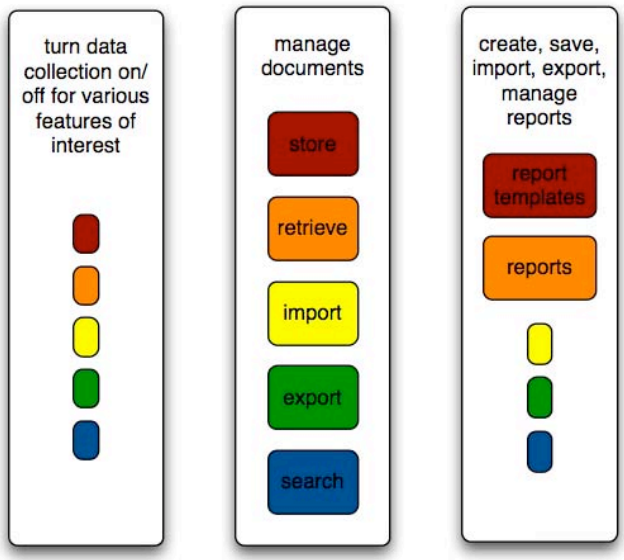

Figure 7: Mental Model Researcher

6.6. Researchers

Researchers also can turn data collection statistics on and off for various features of interest, which may or may not be the same as the staff member features of interest. They can also manage the documents that are in their repository. This includes storing, importing, exporting, retrieving, and searching. Finally researchers can create, save, import, export, and manage their reports as well. Again, this includes regular reports as well as ad-hoc reports.

7. TECHNOLOGIES EMPLOYEED

In order to create the vEOC, we have employed a variety of technologies. One of the most unique aspects is that all development and deployment is being accomplished in virtual machines (VMs) using VMware [24] and Ubuntu [22]. We do this in order to facilitate a faster mean time to repair and to isolate various servers from each other and from the underlying hardware. For example, we have a VM set up for show only. This VM has pure public access and therefore has been stripped of all confidential information. It is located at http://veocblue.cse.nd.edu/RegularLogin.php. In addition, we have a development VM set up. We do all of our development on this machine, which is the machine reserved for verified builds of the prototype. Other technologies employed include XHTML, CSS, Dynamic HTML, AJAX, Reverse AJAX, PHP, JavaScript, MySQL, Jetty and Apache. Finally, we also use Dojo [19] and the Yahoo User Interface (YUI) [23] third party toolkits to create tabs and widgets for the desktop. Figure 8 shows how we link all these technologies together to create the virtual EOC.

The architecture of the vEOC consists of a set of clients connected via the internet to a centralized server. The server is Apache 2 and the Jetty-6.1.7. The web client connects to the servers and requests a HTML/JavasSript file, which it then renders. Any number of clients can connect to the server, limited only by the load on the server. Clients initially render the page using AJAX and then subscribe to a channel to have information concerning the exercise "pushed" to them via Reverse AJAX. In this way, clients finally are able to participate in the exercise with other clients.

7.1. Server Push

In our prototype, we use Reverse AJAX, sometimes also called Comet[16], to implement server push using HTML, AJAX, and JavaScript. Server push is a technology that enables the server to initiate communication to other clients over HTML channels. This differs from standard HTML because in standard HTML, only the client can initiate communication to another client or to a server. We implement this using Jetty-6.1.7[20] and the Dojo toolkit [19].

8. VEOC ARCHITECTURE

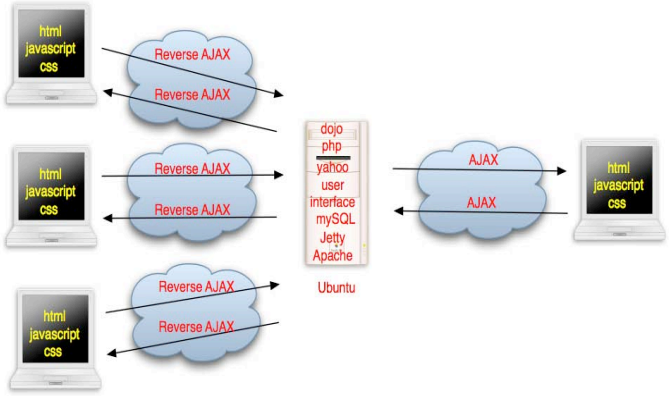

Figure 8: vEOC Architecture

The architecture of the vEOC consists of a set of clients connected via the internet to a centralized server. The server is Apache 2 and the Jetty-6.1.7. The web client connects to the servers and requests a HTML/JavasSript file, which it then renders. Any number of clients can connect to the server, limited only by the load on the server. Clients initially render the page using AJAX and then subscribe to a channel to have information concerning the exercise "pushed" to them via Reverse AJAX. In this way, clients are able to participate in the exercise with other clients.

9. VEOC INFRASTRUCTURE

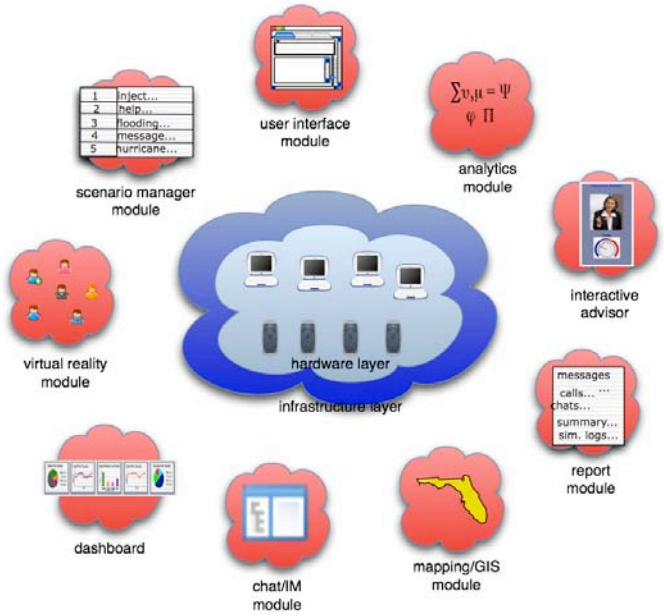

Figure 9: vEOC Infrastructure

The vEOC infrastructure consists of a set of interconnected modules. The modules include a mapping/GIS module, an analytics, chat/IM module(VOIP), scripting module, a user interface module, a dashboard module, a report module, a virtual reality module, and an interactive advisor module. We now explain each module and its corresponding functionality in depth.

9.1. Mapping/GIS Module

The mapping/GIS module is the place in which we incorporate GIS/mapping capabilities. Currently, we do not have an implementation for this.

9.2. Analytics

The analytics module is the place in which we do necessary calculations to support other modules. Currently, we do not have an implementation for this.

9.3. Chat/IM Module,VOIP

The chat/IM module deals with a local chat function. To implement this, we use Reverse AJAX [16] to create a real-time collaboration environment.

9.4. Scripting Module

The scripting module stores the code that deals with implementing automatic sending of injects from the script stored on the server to the clients who are participating in the exercise. It also deals with the user input status and pushing the response of the inject from the single user to the other clients.

9.5. User Interface

The user interface module is the location in which the sections of code reside that deal with the display of information on the server to the user. It is mostly client code, and it consists of HTML and JavaScript.

9.6. Dashboard

The dashboard module is the place in which we implement the dashboard. Currently, we do not have an implementation for this.

9.7. Report Module

The report module is the central location for the code that generates report templates and reports. This also enables one to save, create, import, export, and search through reports.

9.8. Virtual Reality Module

This is a plug-in module that enables one to incorporate virtual reality into the vEOC. We do not have an implementation for this at the moment.

9.9. Interactive Advisor Module

The interactive advisor module is the place in which we implement the interactive advisor. Currently, we are using JESS version 7.1.2 [10]. (See section 10.2.1.3)

10. RESULTS

We combine all of these element using our vEOC architecture, and so far, we have created iteration #2 of the vEOC. Basically, this is an early prototype to get user feedback and a proof-of-concept application that shows how the system looks and works. Now we will discuss screenshots for each of the user views in the vEOC.

10.1. Login

Figure 10 shows a generic user login screen.

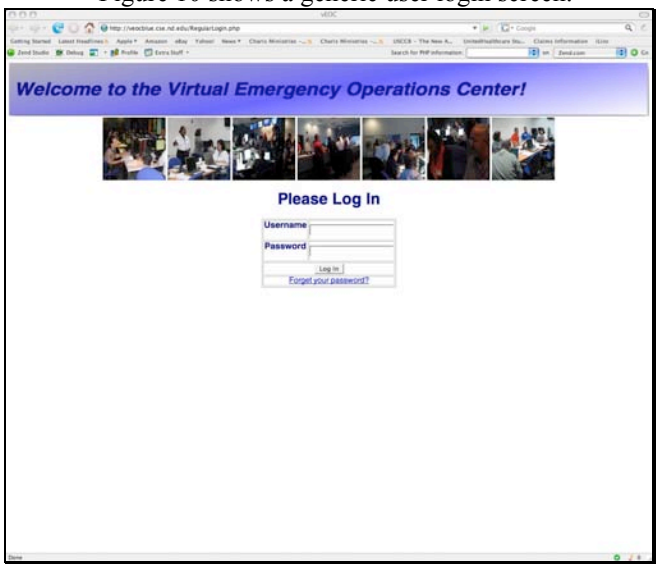

Figure 10: A Generic User Login Screen

Figure 11: A User Role and Script Selection Screen

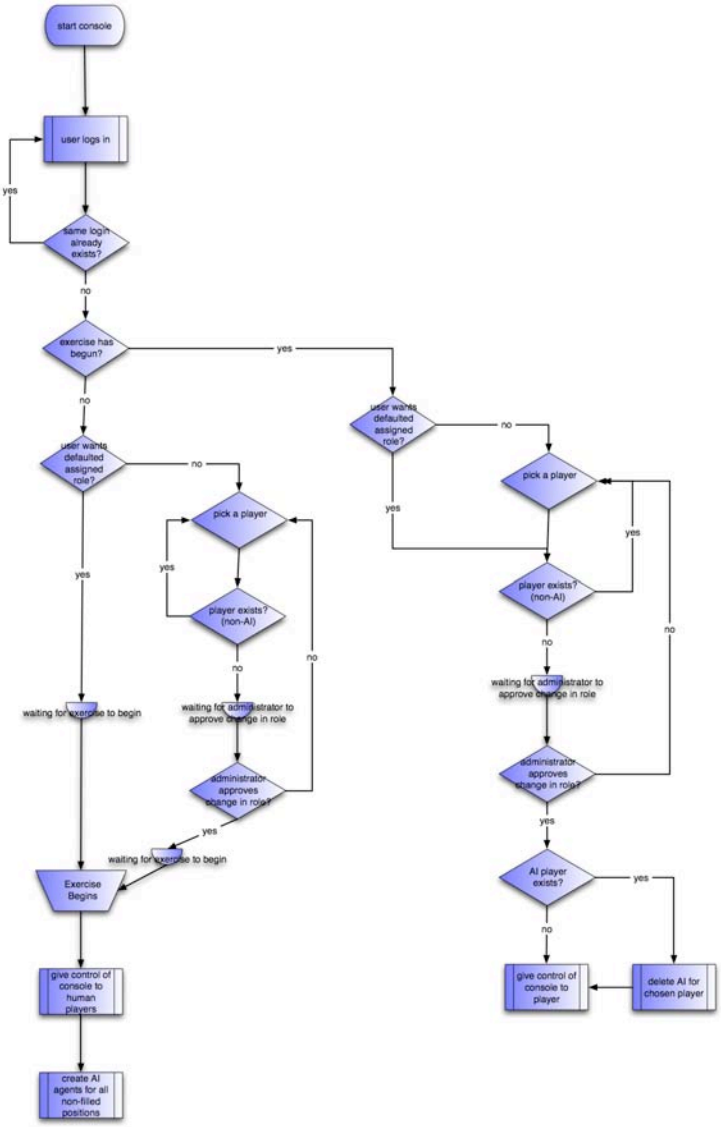

Figure 12. Trainee Startup flow

After the trainee logs in, the user is brought to the role and script selection screen. See figure 11. This is where the user selects the role he/she wants to exercise and the script on which he/she wants to train.

10.1.1. Startup Flow

Figure 12 shows the startup flow that ensues for each trainee. When a trainee logs in, he/she next selects a role and a script that he/she wants to train on. First, we ensure that the same login does not already exist. Next, we check to see if the exercise has begun or not. If the exercise has already begun, then we check to see if the user

wants to use the default assigned role that was assigned to him/her upon creating a login access. If the user wishes to use his/her usual role, then we check to see if an artificial agent (AA) exists or not. If an artificial agent does not exist, then the trainee is now able to participate in the exercise. If an artificial agent does already exist, then we delete the AA, and allow the trainee to join the exercise. If the trainee does not want to use his/her default role, then the trainee attempts to choose another role. If the role is already taken, then the trainee cannot choose this role. Otherwise, the administrator must approve the change in role. Upon approving the change in role, then the AA is deleted for that role (if it exists), and the trainee is allowed to join the exercise. If the exercise has not begun, then there are not artificial agents already instantiated, so the trainee essentially goes through the same process of choosing between his/her default role and a new role with which to train. Finally, the trainee has control of the console (with the exception of reacting to status, since there is no status to react to), and the trainee waits for the exercise to begin. Note that in the last case, the administrator must approve any change in roles by the trainee. This makes sense because personnel usually come to the EOC and are trained to fulfill a particular role. They usually do not train on other roles. For example, the lead police agency representative would not be qualified to be a Parks and Recreation representative as well. In addition, if we let anyone train on any role, the person who logged in first as a particular role would get the role to the exclusion of the proper personnel.

10.2. Trainee

In Figure 13, we see a screenshot of the main trainee desktop. There are three main tabs on the desktop: desktop, history, and sitreps. We now explain each main tab in turn.

10.2.1. Desktop

The Desktop is the main place in which trainees interact with the virtual EOC. The Desktop is further divided into four main sections: communication tools, the console manager, the interactive advisor, and the dashboard.

10.2.1.1. Communication Tools

The communication tools enable the trainee to receive and react to injects. In this section, we simulate various communication tools, including the telephone, the cellphone, face-to-face communication, PDAs, and radios. (See figure 14)

Figure 13: Trainee Desktop

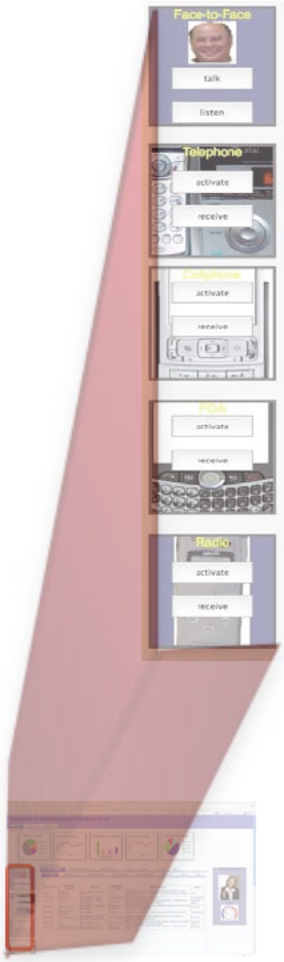

Figure 14: Trainee Communication Tools

10.2.1.2. Console Manager

The console manager enables the trainee to view the current status of the exercise. (see figure 15)

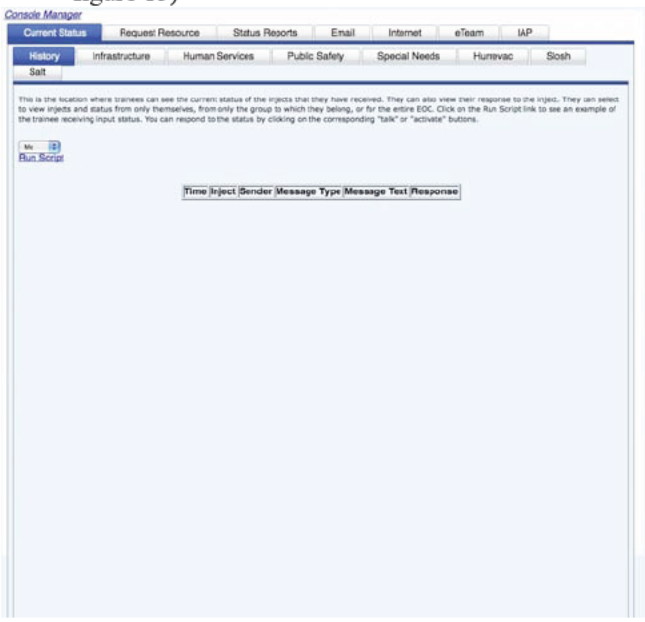

Figure 15: Console Manager

It also is the main location in which the trainee can interact with various third-party tools such as eTEAM, Hurrevac, SALT, and SLOSH. The console manager also includes a place for current status in which the user can view the current Incident Action Plan (IAP). See figure 16.

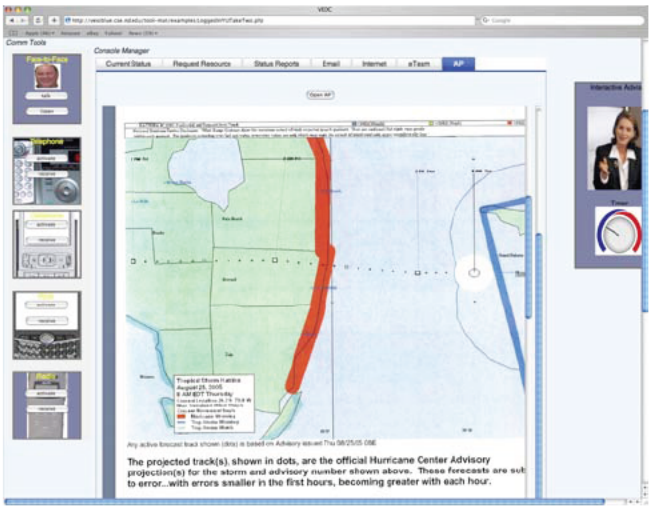

Figure 16: Viewing the Current Incident Action Plan

10.2.1.3. Interactive Advisor

The interactive advisor provides real-time feedback to the trainee based on his/her crisis management decisions. It analyzes the decisions of the trainee and compares the decision to the "correct decision" based upon standard operating procedures (SOPs) and current subject matter expert opinion. It then offers the trainee feedback and advice on his/her decision.

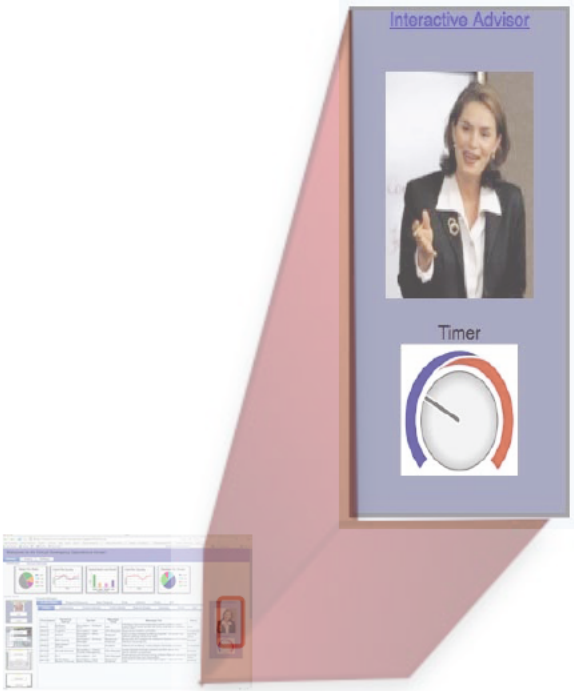

Figure 17: Interactive Advisor

10.2.1.4. Dashboard

The dashboard provides real-time feedback to the trainee concerning the effect of his/her decisions on various aspects of various resources. (see figures 17) For example, one of the categories is water per shelter. If the trainee makes a decision that affects the amount of water per shelter, then the dashboard will update and show a decrease or increase in the amount of water per shelter. This is a feature requested by the crisis decision making researchers to help offer the trainee immediate feedback of the results/impact of his/her decision [14].

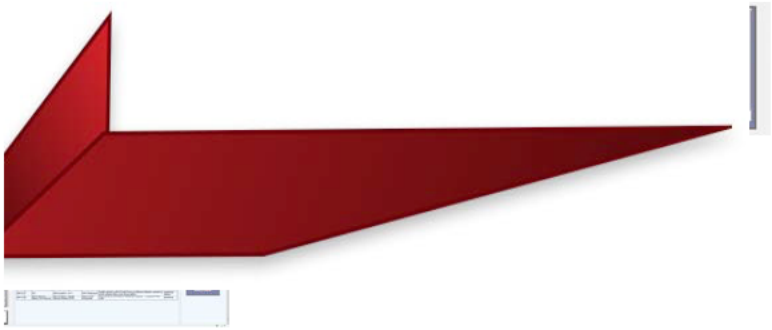

Figure 18: Dashboard

10.2.2. History

This is the place where users can go to view the history of the injects and status that was presented to them. They can also search through the history to find injects of other items of interest in this area. See figure 19.

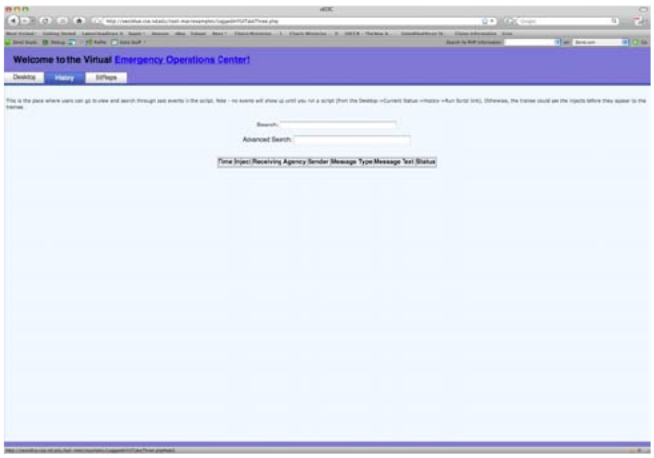

Figure 19: Trainee History Tab

10.2.3. Sitreps

This is the place where users can go to create and view current situational reports (sitreps). Sitreps are used to keep a log of what occurred on a person's shift and also to facilitate a smooth transition during shift changeover. See figure 20

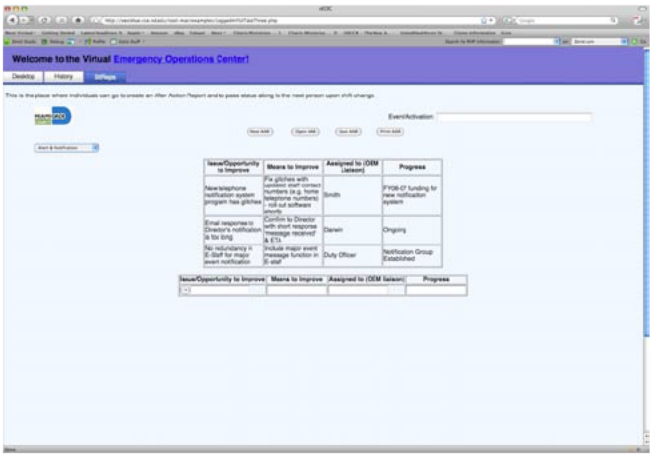

Figure 20: Trainee Sitrep Tab

10.3. Observer

The observer's desktop is the same as the trainees desktop, except that the observer cannot inject status into an exercise.

10.4. Scenario Manager

The Desktop is the main place in which scenario manager interacts with the virtual EOC. There are three main tabs on the desktop: desktop, script manager, and library. We now explain each section in turn.

10.4.1. Desktop

The Desktop is the main place in which scenario manager manages the exercise. There are 6 main controls that the scenario manager can manipulate: start, stop, resume, pause, next block, and fasttime. See figure 21. We now explain each control in turn.

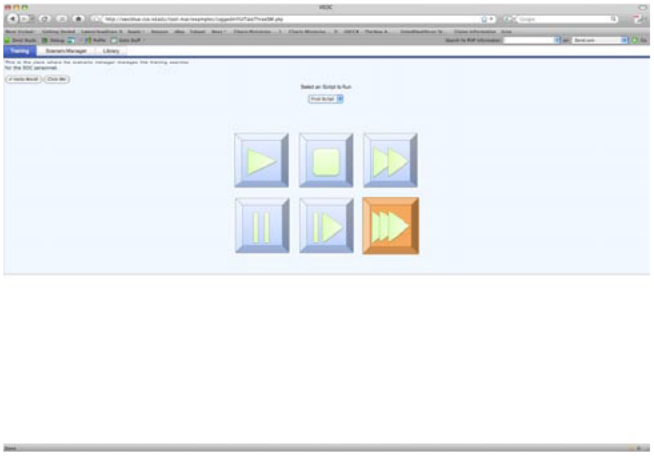

Figure 21: Scenario Manager Desktop Tab

10.4.1.1. Start

This control begins the exercise.

10.4.1.2. Stop

This control terminates the exercise

10.4.1.3. Resume

This control resumes the exercise from the last point before it was paused.

10.4.1.4. Pause

This control temporarily suspends exercise execution.

10.4.1.5. Next Block

This control sends the next inject to the trainee.

10.4.1.6. Fast Time

This control speeds up the internal clock of the exercise, and it allows the exercise to speed up. Clicking on this button again will cause the clock to resume normal time.

10.4.2. Script Manager

This is the place where the scenario manager creates and edits training scripts. See figure 22

Time	Inject	Receiving Agency	Sender	Message Type	Comm Tool	Message Text	EXPLAN OBJ
0000	1	M Beach Divisional	Simulation - M Beach EOC	Info	telephone	City Mayor and commissioners want to confirm when topical storm force winds will arrive and which shelters will be open.	1,4
0002	2	Logistics	Simulation - State	Info Request	telephone	State wants location of PODs.	2
0003	3	AHCA	Simulation - Mercy Hospital	Request	pda	8 ICU nurses needed at Mercy Hospital hospital has difficulty getting additional staff	2,4
0005	4	MD Housing	Simulation - Building Manager	Resource Request	radio	Buses with AC needed for displaced residents at Culmer Gardens Housing fire.	1,4
0007	5	Animal Services	Simulation - Animal Shelter Management	Info Request	cellphone	Animal Shelter Director wants to confirm when is the animal shelter opening.	1,3
0009	6	311	Simulation - 311	Info Request	face-to-face	Foster family with 5 kids living on Miami Beach want to know where they can evacuate to.	2,3
Time	Inject	Receiving Agency	Sender	Message Type	Comm Tool	Message Text	EXPLAN OBJ

Figure 22: Script Manager Tab

10.4.3. Library

This is the place where the scenario manager stores previous and current scripts. See figure 23

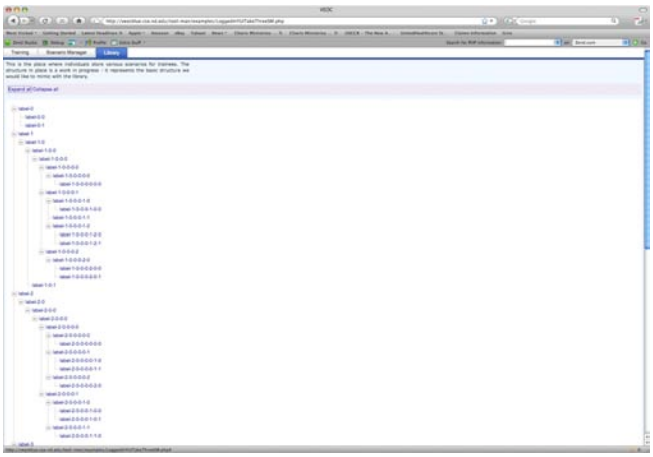

Figure 23: Scenario Manager Library Tab

10.5. Staff Member

The staff member console is the main place in which staff members interact with the vEOC. There are four main tabs on this console: the newsroom, data collection, reports, and documents. We now explain each section in turn.

10.5.1. Newsroom

The newsroom is the main place in which staff members interact with each other in the virtual EOC. The newsroom has several composite features: chat/instant message (IM), a calendar, a news bulletin, a search, and advanced search feature. See figure 24

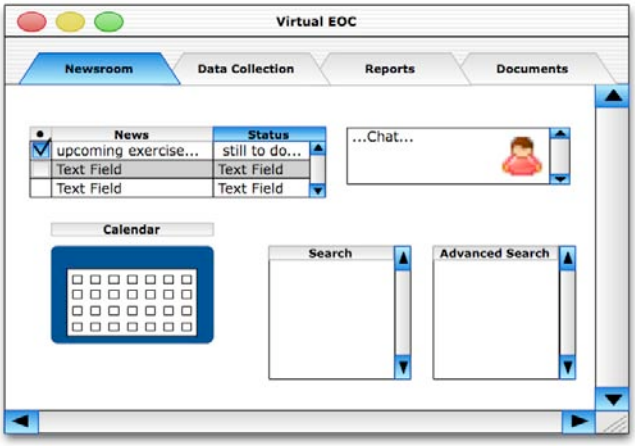

Figure 24: Staff Member Newsroom

10.5.2. Data Collection

This is the place where researchers can turn on or off data collection for various features of interest. See figure 25

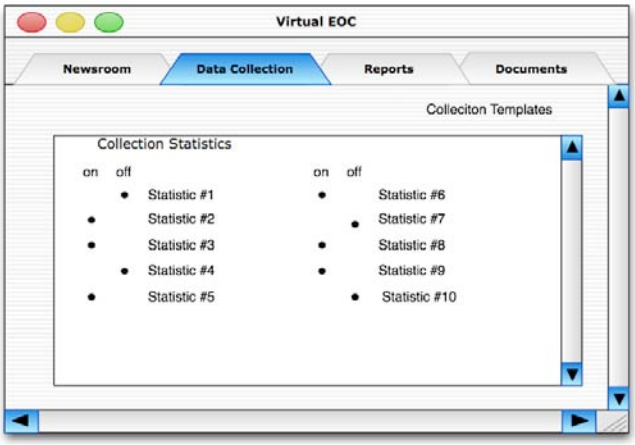

Figure 25: Staff Member Data Collection Tab

10.5.3. Documents

This is the place where staff member can create, store, and retrieve common documents. See figure 26.

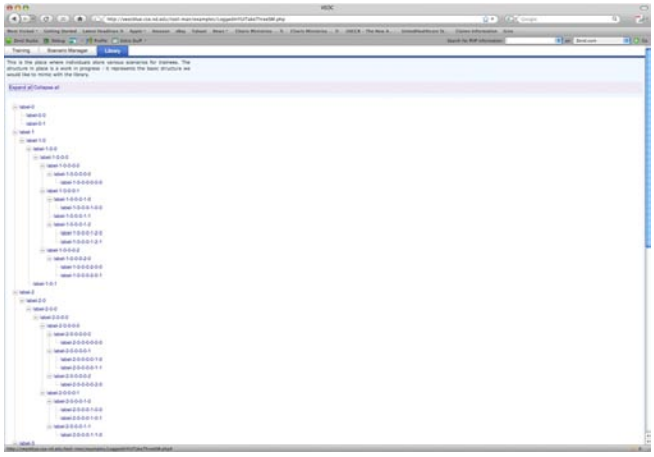

Figure 26: Staff Member Documents Tab

10.5.4. Reports

This is the place where staff members can create, store, and retrieve common reports and report templates. Report templates are pre-formatted documents that are used to create frequently-used reports. See figure 27.

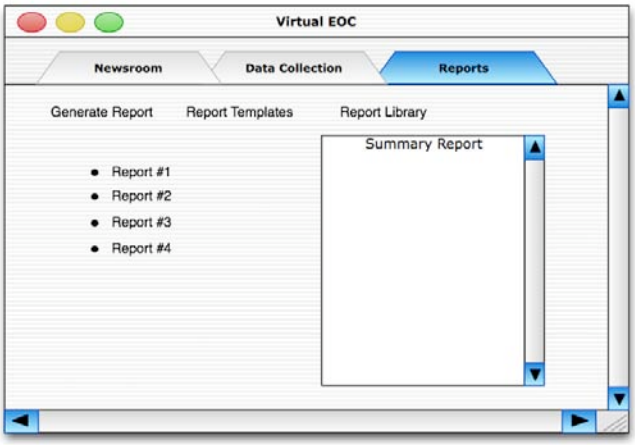

Figure 27: Staff Member Report Tab

10.6. Administrator

The administrator console is the main place in which administrators interact with the vEOC. There is one main tab on this console: the user profile tab.

10.6.1. User Profile Tab

The user profile tab is the main place in which administrators interact with the virtual EOC. Administrators have the ability to create, edit, and delete profiles, virtualize profiles (that is, supplant humans with a corresponding artificial agent), and devirtualize profiles (that is, supplant artificial agants with their corresponding human counterpart) Administrators also provide user access control through this console. See figure 28.

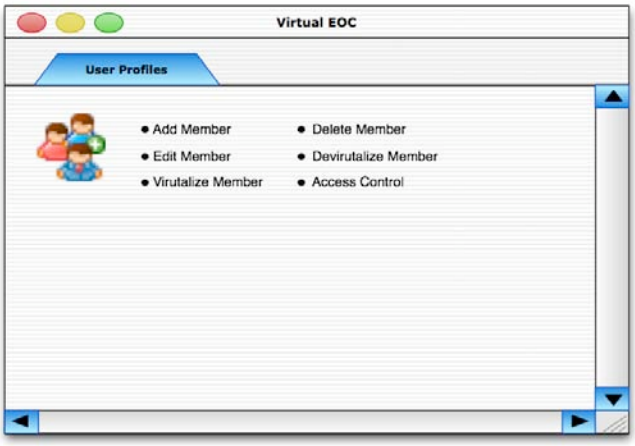

Figure 28: Administrator User Profile Tab

10.7. Researcher

The researcher console is the main place in which researchers interact with the vEOC. There are four main tabs on this console: the newsroom, data collection, reports, and documents. We now explain each section in turn.

10.7.1. Newsroom

The newsroom is the main place in which researchers interact with the virtual EOC. It is similar to the staff member newsroom in terms of its functionality and design.

10.7.2. Data Collection

This is the place where researchers can turn on or off data collection for various features of interest. It is similar to the staff member data collection tab in terms of its functionality and design.

10.7.3. Reports

This is the place where researchers create, store, and retrieve common reports and report templates. It is similar to the staff member reports tab in terms of its functionality and design.

10.7.4. Documents

This is the place where researchers create, store, and retrieve common documents. It is similar to the staff member documents tab in terms of its functionality and design.

11. KEY FEATURES

There are several key features of the vEOC that are worth highlighting.

11.1.1. Secure Login

To ensure secure login, we use the Lightweight Directory Access Protocol (LDAP). This is an application protocol for securely accessing directories over TCP/IP.

11.1.2. Different Login Interfaces Depending On Which Role You Are

At the Miami-Dade EOC, different users have a need for different kinds of information depending on who the user is and what role the user exercises. For example, not everyone should be privy to Hurrevac results, lest they try to make judgments they are not authorized to make based on this information. [13] When the user logs in, only certain users are able to access the Hurrevac plug-in. Other users will not.

11.1.3. Automated Scripting

The computer automatically sends injects to the client based on the script that is input from the scenario manager. The user then interacts with the injects. The user receives status via the communication tools, and the user reacts to the status again through the communication tools.

11.1.4. Multi-browser Compliance

This prototype is compatible with Internet Explorer (IE) 7.0 as well as Firefox 2.0.18 and 3.0.4, and Opera 2.0.4.

11.1.5. Multimedia

Currently, we have audio support for receiving injects. In the future we also want to include interactive inject video support as well.

12. FUTURE WORK

Our next step is to begin implementing the interactive advisor and the dashboard. After that, we will try an experiment with a group of students to test the software.

13. ACKNOWLEDGEMENTS

We thank the Miami-Dade EOC for sharing their time and insights with us throughout this project, especially Mr. David Perez, Mr. Frank Reddish, and the Logistics Section.

14. CONCLUSION

In this work, we have introduced our vEOC prototype. We began with a short introduction and rational for computer-based solutions to training upper-level emergency managers over traditional face-to-face solutions. Next we introduced our vEOC prototype: project Ensayo. We discussed our main research and design methodologies. We also discussed the 6 different user views in depth, included each user's mental model. Following that, we discussed the various technologies that link together to form the vEOC. We then delved into the architecture and infrastructure behind the vEOC. After that we described the basic user interfaces for each of the 6 different user views. Finally, we concluded with future directions of the virtual EOC prototype.

References

1. Sniezek J., D. Wilkins, P. Wadlington, and M. Baumann. Training for Crisis Decision-Making: Psychological Issues and Computer-Based Solutions. *Journal of Management Information Systems*. Vol. 18, No. 4, Spring 2002, pp. 147 – 168.
2. Green, W. "E-emergency Management in the U.S.A.: A Preliminary Survey of the Operational State of the Art. *International Journal of Emergency Management*. Vol. 1, No. 1, 2001, pp70-81.

3. Agrait, R., A, English, D. Evans, T. Hammell, J. Loughran, and M. Stahl. Review of Models, Simulations, and Games for Domestic Preparedness Training and Exercising: Volume III. *Office for Domestic Preparedness. Department of Homeland Security*. March 2004.
4. Liu, D., T. Nickens, and Y. Wang. Modeling and Simulating Decision-Making Learning Process under Crisis Situation. *Proceedings of SISO 2006 Fall Simulation Interoperability Workshop*, Orlando, FL, 2006.
5. Statement of N. Rabkin. Hurricane Katrina: Providing Oversight of the Nation's Preparedness, Response, and Recovery Activities. *GAO Testimony Before the Subcommittee on Oversight and Investigations, Committee on Energy and Commerce, House of Representatives*. September 2005.
6. Boehm, B. A Spiral Model of Software Development and Enhancement. *IEEE*, 21(5):61-72, May 1988
7. Sebesta, R. Programming the World Wide Web. Pearson Education, Inc, 2008.
8. Young, I. Mental Models. Aligning Design Strategy With Human Behavior. Rosenfeld Media, 2008.
9. Clark, C. CAPP 30523 Lecture Notes. Fall 2008.
10. JESS toolkit. Available at http://www.jessrules.com/jess/index.shtml.
11. Becerra-Fernandez, I. Prietula, M., Madey, G. , and D. Rodriguez. Project Ensayo: a Virtual Emergency Operations Center for Disaster Management Research, Training, and Discovery. Presentation and Proceedings, The First International Conference on Global Defense and Business Continuity (ICGD&BC 2007). San Jose, CA. July 2007.
12. Miami-Dade Exercise EOC Activation. June 2008.
13. Perez, D. Conversation With The Author. 14 August 2008.
14. Miami-Dade EOC Activation EOC. September 2008.
15. Perez, D. Conversation With The Author. 2 September 2008.
16. Crane, D and P McCarthy. Comet and Reverse Ajax. Springer-Verlag New York Inc. 2008.
17. Russell, M. Dojo: The Definitive Guide. O'Reilly Media, Incorporated, June 2008.
18. Miami-Dade EOC Training Script. Miami-Dade Exercise EOC Activation. June 2008.
19. DOJO toolkit. Available at http://dojotoolkit.org.
20. Jetty Server. Available at http://www.mortbay.org/jetty.
21. Apache Server. Available at http://httpd.apache.org.
22. Ubuntu Operating System. Available at http://www.ubuntu.com.
23. Yahoo User Interface Toolkit. Available at http://developer.yahoo.com/yui/.
24. VMware server. Available at http://www.vmware.com/.