Graphical Interfaces for a Large-scale Human Command in Emergency Response

A Project

Presented to

the Faculty of the School of Engineering and Applied Science

University of Virginia

In Partial Fulfillment

of the Requirements for the Degree

Master of Computer Science

by

Lori E. Stotler

April 2004

Lori E. Stotler
lstotler@cs.virginia.edu

264 Colonnade Drive #21
Charlottesville, VA 22903
434-825-0110

15 Crestview Lane
Fairmont, WV 26554
304-363-5868

Graphical Interfaces for a Large-scale Human Command in Emergency Response

Graphical Interfaces for a Large-scale Human Command in Emergency Response

Lori Stotler

Abstract

Recent terrorist attacks have demonstrated weaknesses in capabilities for detecting coordinated terrorist attacks and responding to them. This paper addresses the System for Terrorism Intervention and Large-scale Teamwork (STILT), a system for enacting a control loop for effective communication and response during emergencies. In particular, this paper focuses on STILT's graphical user interfaces for human interaction. For my project, I have developed three interfaces for use in STILT. First, Wiglaf is a command reception and event generation interface. It is meant to be carried by responders, so that they can receive commands during an emergency and report feedback. Next, Lucca and the Operational Area Map interface aid in the planning and direction of response. The Operational Area Map gives decision makers a view of the emergency situation, while Lucca gives them the ability to send commands to responders.

1 Introduction

In recent years, acts of terrorism have necessitated a critical look at capabilities for detecting coordinated terrorist attacks and responding to them. Part of this analysis has focused on communication for emergency analysis, information sharing, coordination, and response. When attacks and disasters occur, the effectiveness of communication has tremendous impact on the success of the response. Unfortunately, a multitude of communication problems arise during a crisis. A few examples are:

- **Lack of information exchange**. The McKinsey Lessons Learned report on the World Trade Center states that the New York Fire Department "lacks the ability to ensure that these bureaus exchange information effectively." The report includes the recommendation to improve the department's ability to receive and distribute information about critical incidents [7].
- **Lack of coordination among different agencies**. During disasters, concerned agencies respond differently, because of their different jurisdictions, levels of preparedness, responsibilities, political considerations, etc. Fire departments, law enforcement, and government officials are drawn into the crisis at different times with dissimilar perspectives. Besides local personnel, some emergencies will involve state and federal personnel and resources. It is essential, but difficult, to coordinate the divergent responsibilities and efforts [1].
- **Lack of authority**. When various organizations respond to an emergency, there may be overlapping authority or no clear authority [1].
- **Lack of control**. During the September 11 World Trade Center and Pentagon attacks, there were problems with "self-dispatching." Organizations, response units, and individuals proceeded on their own initiative directly to the disaster site without the permission of commanders. These responders included fire fighters who were off-duty, port authority police officers who left their posts at bridges and tunnels, and ambulance crews who converged on Manhattan. Meanwhile, at one point the city had no ambulances available for approximately 400 non-Trade Center emergency calls [3]. This loss of control complicated command at the scene and increased the risks to the responders who had officially been called to action.

In an ideal situation, these issues would not be present. Information would be distributed quickly to those who would use it to plan a response. Individuals would then be directed to enact the response or to continue their normal responsibilities. Feedback would be collected from responders so that the response could be assessed and modified. This type of control loop is currently difficult to enact within a single area and is infeasible at a larger scale.

The System for Terrorism Intervention and Large-scale Teamwork (STILT) was built to provide the interfaces and communications system for enacting a control loop as described above. STILT is built as a specific instantiation of the Willow architecture, a system originally designed for the management of large, distributed information systems. The Willow system is made up of two primary components. First, an analysis engine correlates events and determines response. Second, a scalable and efficient communications system and distributed workflow system enact the response. STILT uses these components to detect and respond to coordinated terrorist attacks.

For my project, I have designed three interfaces which apply the Willow command paradigm to the STILT human system. These interfaces are (1) Wiglaf, a command reception and event generation interface, (2) Lucca, a workflow injection interface, and (3) Operational Area Map interface, an interface for the display of sensor events. In this paper, I will give an overview of the Willow system, because STILT is constructed as a specific instantiation of the generic configurable Willow architecture implementation. I will then describe STILT in greater detail. Finally, I will describe the Wiglaf, Lucca, and the Operational Area Map interfaces.

2 Willow

The Willow system is designed to support the survivability of large, distributed information systems [5]. A survivable information system has facilities to provide one or more alternate services in a given operating environment. These alternate services may be different, less dependable, or degraded compared to normal service. An alternate service is required in the event that the system's full service cannot be provided because of failures. Multiple alternate services should be available, so that the system can cope with different forms of damage, damage that occurs under different circumstances, and damage that worsens over time [6].

As part of its survivability approach, Willow uses a range of mechanisms to deal with the faults of information systems, applying fault avoidance, fault elimination, and fault tolerance in one integrated system [5]. For instance, fault avoidance is achieved by disabling vulnerable network elements intentionally when a threat is detected or predicted. Fault elimination is realized by replacing system software elements when faults are discovered. Lastly, fault tolerance is attained by reconfiguring the system when non-maskable damage occurs [2].

The Willow architecture is an extended form of what is known as a survivability architecture or information survivability control system [9]. The key to the architecture is a control loop which monitors the information system, analyzes sensor data to identify errors, and effects responses to identified problems.

Willow provides automated fault tolerance over large, complex systems with an innovative distributed architecture, shown in Figure 1. Its major components are formal specifications, Siena, Selective Notification, Freeflow, ANDREA, SPARTAN, and TEDL.

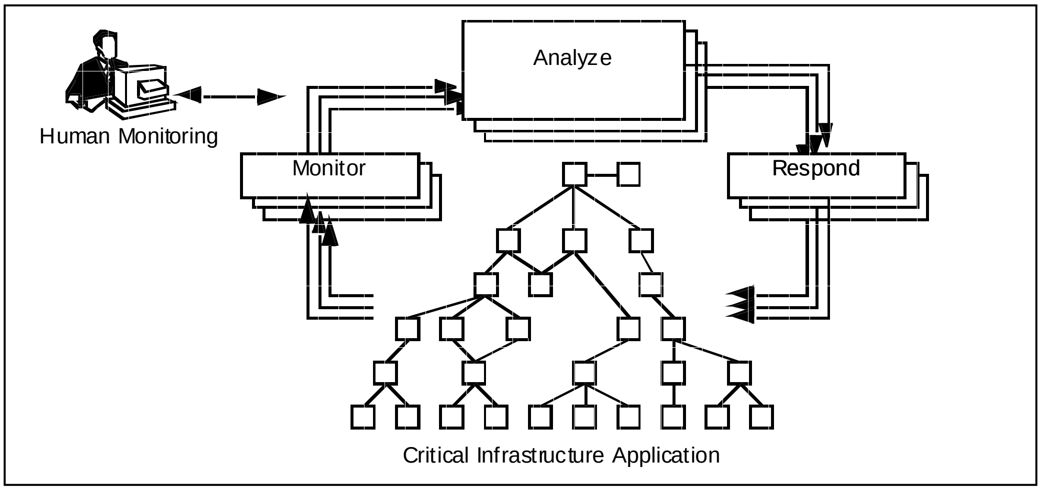


Figure 1: A high-level overview of the Willow Reactive Survivability Architecture

Formal Specifications

Analysis and response in Willow is driven by formal specification. A series of specifications of fault detection and reaction detail the exact behavior of the system for a given application and are used to generate a specific instantiation of Willow. All primary components of Willow are directly synthesized and driven from these specifications [11].

Siena

Siena is a content-based publish/subscribe event notification service. It provides scaleable and efficient networking [10].

Selective Notification

Selective Notification is a communication mechanism that provides symmetrically addressable, decoupled event dissemination. Communication typically involves three primary parts: a sender, a message, and a receiver; however, clients can interact without having knowledge of one another [8]. Messages are targeted based on the properties of the sender, the message, and the receiver. Senders and receivers both expose an “antigen” of their local properties (i.e. rank and location). This allows receivers to specify which senders should be allowed to send them messages and senders to select the receivers of a message. Messages are therefore targeted to only relevant receivers [11].

Freeflow

Freeflow is a workflow system and language that allows for the creation of tasks [2]. Tasks may describe constraints over the system configuration or processes that must be executed. Tasks are hierarchical. They may contain subtasks and can be ordered. Each workflow task contains:

- **An intention**: An intent is the declaration of the constraint to be enforced or the process to execute. Intentions are predefined, enumerable, and hierarchical. Intentions in the same tree are considered to conflict and therefore require resolution through their reasons.
- **A reason**: A task’s reason is a statement of the purpose behind a constraint or process. Each possible reason is a member of a totally-ordered set of reasons that define the priority of tasks. Tasks with lower reasons should be preempted by those with higher reasons. [11].

ANDREA

The ANDREA management service combines the features of Selective Notification and Freeflow. It takes as input hierarchically decomposable workflows consisting of a set of partially-ordered tasks. Selection "functions" describe by property who should receive tasks. Tasks are then projected using Selective Notification to the relevant sites, and receivers attempt to either enact the constraints or perform the tasks. Scalable feedback about the workflow's execution or enforcement is provided to the task's submitter [11].

SPARTAN and TEDL

Detection in Willow is performed by the Time-based Event Detection Language (TEDL) and the System for Proactive, Adaptable, Reconfigurable, and Tolerant Application Networks (SPARTAN). TEDL is an object-oriented formal specification language designed to specify the symptoms of the error states of a system. These formal specifications are then automatically translated into executable implementations of SPARTAN. SPARTAN is a hierarchical, adaptive control system to detect and respond to network sensor events. It is arranged as a group of distributed detectors. Events are detected at the appropriate level of locality and severity in a hierarchy of detectors. Various sensors input timestamped events containing state information from the source. A detector machine generates an output if it enters a state designated as requiring action, in the form of higher-level events or system constraints. System constraints are emitted as workflows with ANDREA [11].

Relevance of Willow to Emergency Management

The chief challenges managed by Willow are scalability and complexity. The information systems of interest are complex arrangements of very large numbers (hundreds of thousands to millions) of computing nodes. They apply extensive communication facilities and operate across many domains of ownership. Thus, the damage to these systems may be complex and widespread and may require complicated error recovery. The Willow system deals with these issues of scalability and complexity in both detection and response.

The problems of scale and complexity are also encountered in emergency management situations. Because Willow is built to be very general and instantiated with formal specifications, it can naturally be applied to the issues of detection and response during attacks and disasters.

3 STILT

STILT, a specific instantiation of the Willow architecture, was primarily built because of the need to detect and respond to geographically-distributed, coordinated terrorist attacks. The system, nonetheless, is general enough that it can be applied to any emergency response situation. A fundamental goal of the STILT project is to create a system that can correlate distributed attacks and precisely target a response. Such a system will increase response effectiveness and will prevent, or minimize the effects of, subsequent attacks [11].

The particular problem addressed by STILT is that of scale. Most emergency response systems are designed specifically for relatively minor, everyday emergencies, such as individual medical emergencies. These are characterized by only having local effects and are usually independent by locality. There is no unified, nation-wide system for information gathering or distribution. STILT provides this using Willow's rich communication mechanism. Currently, the three primary communication tools used in emergency response are (1) two-way radio, (2) mobile telephone, and (3) mobile data terminals. The two main modes of communication are unicast and broadcast. The communication facilities available through STILT, on the other hand, provide a rich model for targeting messages only to relevant parties based on the properties of the sender, the message, and the receiver. STILT's notification targeting capabilities are flexible such that the same interfaces can be used for both wide-area detection and response for "on-scene" command and coordination of large groups of responders [11].

The core Willow implementation has been adapted for use in emergency response simply by specifying how the system should behave and creating the necessary graphical interfaces for human use. STILT consists of (1) the core Willow software infrastructure, (2) a collection of formal specifications describing the specific behavior of Willow, and (3) a collection of graphical user interfaces for human interaction [11].

STILT interfaces

As discussed above, at the heart of the Willow system is idea that analysis and response should be driven from formal specifications. These specifications are then used to generate a specialized instantiation of the general Willow system, such as STILT. These specifications are additionally used to create the STILT graphical user interfaces. Consequently, it is possible to easily change the specifications, for instance to add new scenarios or add detail, and thereby automatically change both the STILT system and user interfaces.

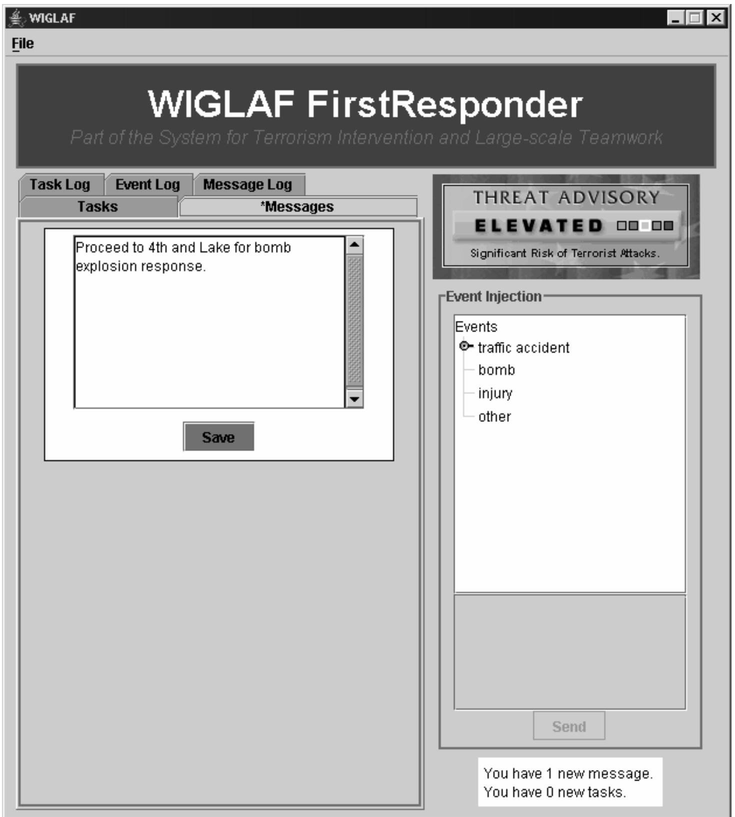
Various interfaces for STILT user interaction have been developed. They are described in the following sections.

4 Wiglaf

Wiglaf is an interface for command reception and event generation. Wiglaf graphical user interfaces are meant to be used by a wide variety of individuals, so that they may carry out tasks and report events in their vicinity.

The central design principal behind the Wiglaf interface is that it must be simple to use. This is necessary for three reasons. First, the interfaces must function effectively in the high-stress, critical environment of a disaster. Second, they are primarily meant for secondary use. STILT devices, for instance, should not distract an EMT from her primary responsibilities, such as treating patients, etc. Third, the interfaces may be used in devices with limited display capabilities; therefore, components must be kept to a minimum. Wiglaf interfaces are additionally designed to be extremely flexible, because they could potentially be used by a wide variety of individuals, such as first responders, hospital workers, government officials, and airport workers. To accommodate this diversity, Wiglaf is designed as a very general, yet flexible interface that can be personalized. Information for personalization is contained both in a configuration file and in the STILT formal specifications.

There are currently two classes of Wiglaf interfaces, tablet and PDA. The tablet interface, displayed in Figure 2, is designed to be run on a tablet, laptop, desktop computer, or mobile data terminal. The PDA interface, shown in Figure 3, is designed to be compact for display on a PDA. It, however, is not currently run on actual PDAs due to technological limitations. The primary difficulty is that only early versions of Java are presently available on PDAs, and Wiglaf requires Java 1.4. (It should be noted that another STILT interface, not developed for this project, is available on PDAs through a web browser.) Tablet interfaces will most likely be used by responders in the field with mobile PCs or by responders who are managing others, while PDAs will probably be used by responders in the field who are performing actionable tasks.



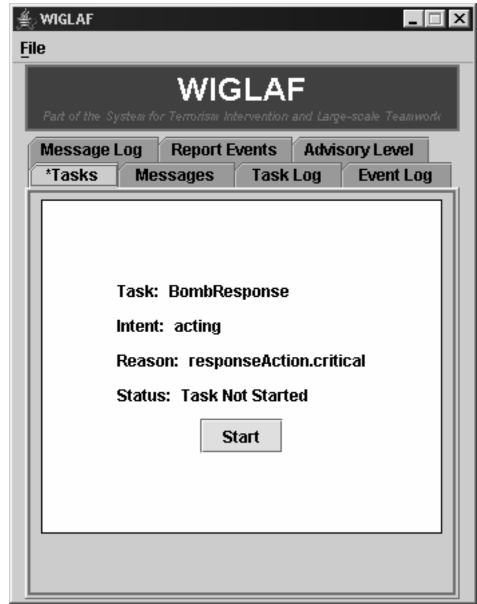


Figure 2: Wiglaf Tablet Interface

Figure 3: Wiglaf PDA Interface

Using Wiglaf, individuals may receive either commands or messages. When a command is received, the user must report her status in completing it. Specifically, the user must indicate when the command has been started, paused, completed successfully, or completed unsuccessfully.

Commands in STILT are transmitted using Willow's selective notification mechanism. Therefore, commands may reach an individual based on attributes of the receiver, sender, and message. Wiglaf users are responsible for setting their attributes and setting "sender qualification" attributes. For example, an Emergency Medical Technician might set her name as "Jane Doe" and her branch as "EMT." She might indicate that she has a defibrillator, and she can set her location using coordinates found with a GPS receiver. She might also specify that individuals sending her commands must be EMT of higher rank. This is accomplished using Wiglaf's File/View Attributes, displayed in Figure 4, and File/Sender Qualification menu options. Note that these attribute settings remain distributed and are not stored in any central repository. In addition, these settings will eventually include some enforcement of correctness by an unimplemented Willow trust mediation component. Individuals will not be able to set all attributes at will. Jane, for instance, cannot declare herself to be the division chief.

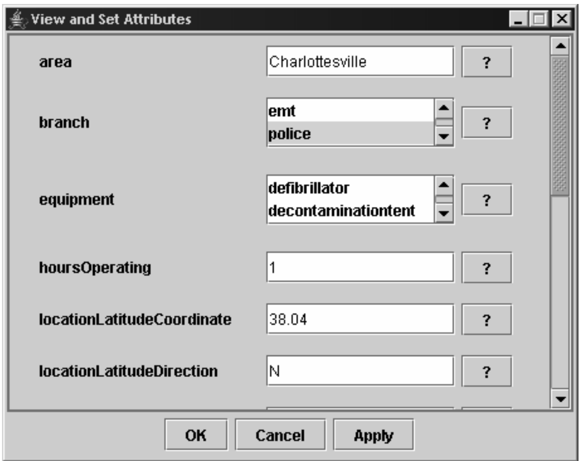


Figure 4: Setting User Attributes

Wiglaf allows users to act as "human sensors" in the STILT instantiation of Willow. Included in the interface is an "Event Injection" panel. Using this panel, users may report events from their environment. These events and the typical event attributes are predefined in formal specifications.

5 Lucca

Lucca, shown in Figure 5, is an interface for command injection. It allows a decision maker to compose tasks, constraints, messages, and complex workflows. Additionally, users can submit their composed workflows to the STILT system and then monitor the command's progress.

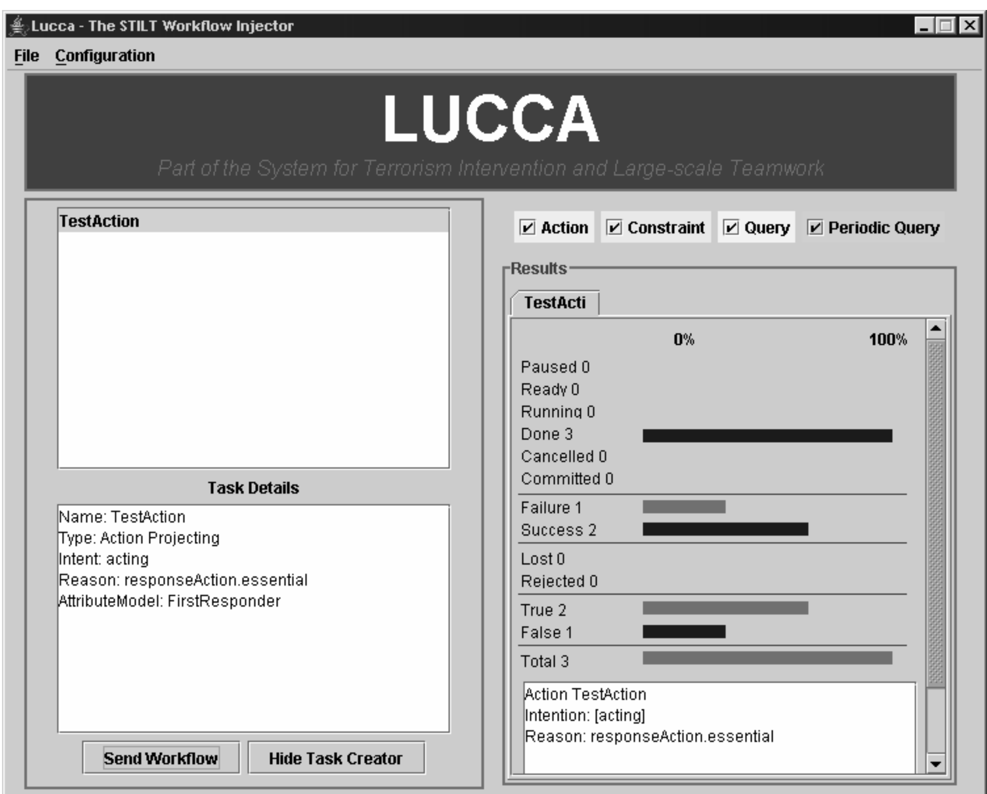


Figure 5: Lucca

The Lucca Task Creator interface, presented in Figure 6, allows a user to create workflows. First, the user must enter basic information, such as the workflow name, type, intention, and reason. Furthermore, the workflow may contain receiver selection constraints for use with Selective Notification when the workflow is delivered. Lastly, the user may set the child tasks of a workflow. Thus, Lucca allows the user to specify and modify very complex workflows. Many of the user choices, such as intention, reason, and receiver selection options, are taken from STILT formal specifications. The resulting interface is powerful and formal, rather than complex and prepared in an ad-hoc manner.

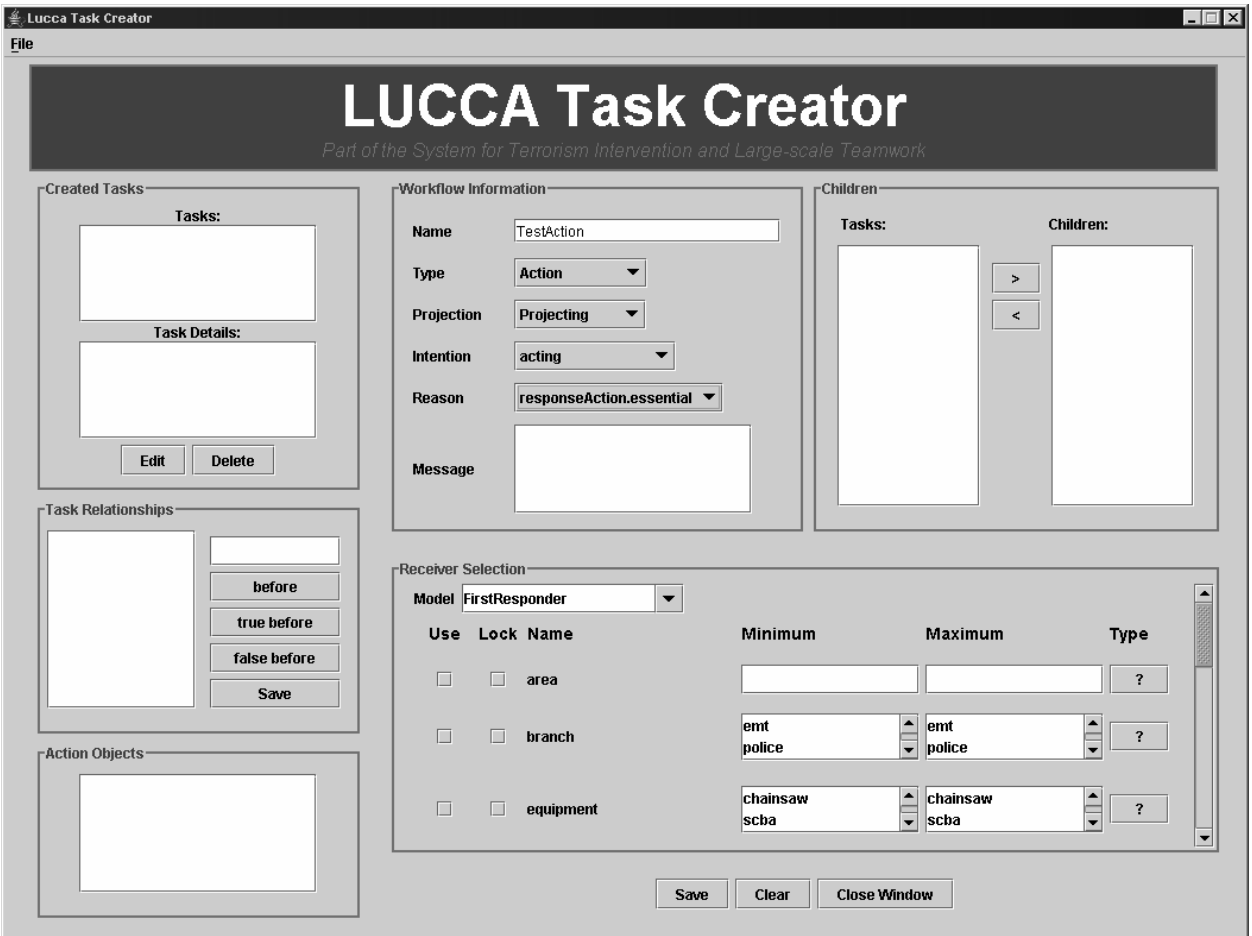


Figure 6: Lucca Task Creation Interface

Besides creating workflows, the user may use the Lucca Task Creator to specify relationships among tasks. For example, a decision maker may specify "Task 1 before Task 2." The tasks in a relationship may be added as children to a workflow.

In realistic use, there may be many tasks which are common and would require frequent regeneration. Therefore, the Lucca Task Creator includes an option for loading Action Objects. Action Objects are prewritten classes, each which run a workflow. These Action Objects can be generated from specifications written in A Workflow Language (AWOL). AWOL is a formal specification language for describing hierarchic selective workflows.

As mentioned above, Willow projects workflows using Selective Notification. Receivers of Lucca workflows have the option of setting “sender qualifications.” Consequently, Lucca contains the option to set the commander’s attributes exactly as Wiglaf attributes are set in Figure 4.

Once workflows are created and/or Action Objects are loaded, the workflows can be submitted to the STILT system. The status of the workflow is displayed in the Lucca Results panel.

6 Operational Area Map

The Operational Area Map (OAM) interface is meant to give decision makers a picture of a particular operational area and the events occurring at a given time. These events might include both the events submitted by Wiglaf users and automated sensor events. The interface can be used in conjunction with Lucca, so that a situation can be analyzed and an appropriate response can be projected.

The primary OAM interface components, as shown in Figure 7, are:

- A map rendered using Microsoft MapPoint Web Services
- A “find” area that allows the user to enter a city or zip code for which a new map will be rendered
- A legend displaying a list of event types which might be received and the corresponding icon which will be placed on the map to represent the event
- A “Send to Lucca” area which allows the user to select an area on the map and then send the coordinates to Lucca for use in receiver selection.

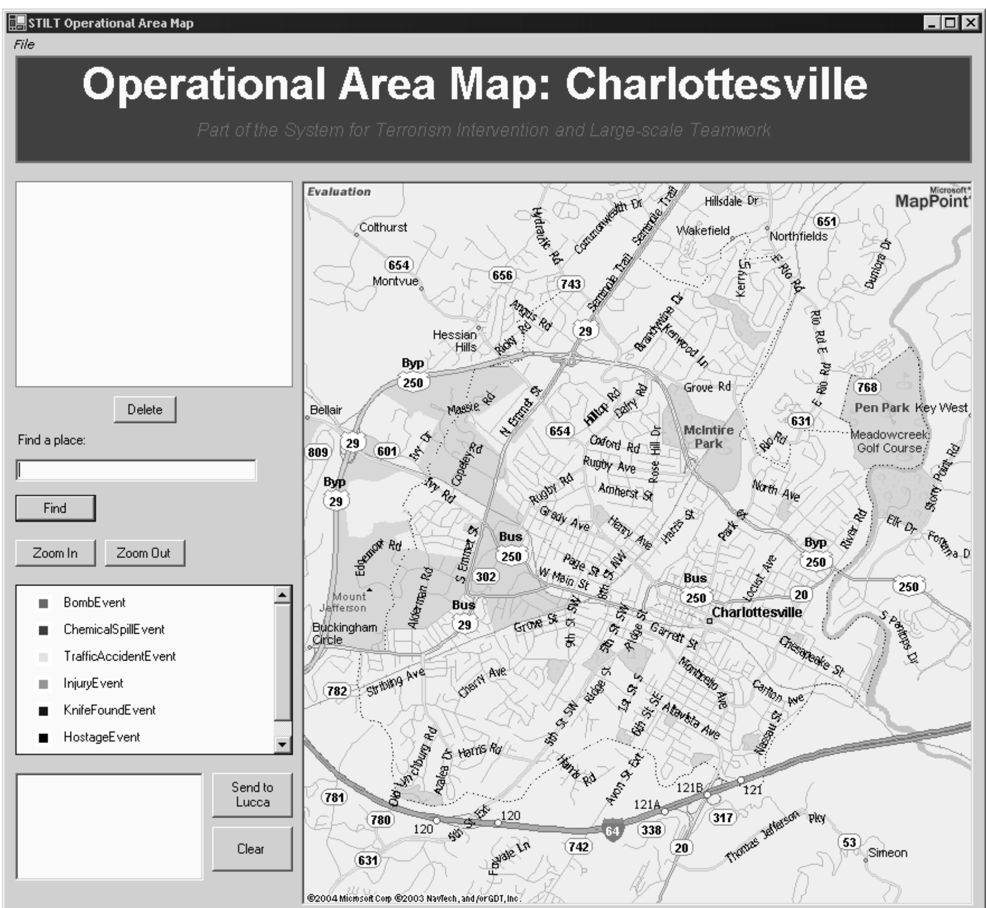


Figure 7: Operational Area Map Interface

As with the other STILT user interfaces, the OAM interface is adaptable. The events and corresponding icons are contained in a simple configuration file.

As mentioned in section 5, the OAM interface contains a "Send to Lucca" feature. At any time, a commander may select a map area by dragging the mouse over the map. This action draws a rectangle on the map, as is shown in Figure 8. The range of selected latitude and longitude coordinates are displayed, and the user may send these coordinates to Lucca. This feature is useful when a task or constraint must be targeted to a specific area in order to respond to an event or events.

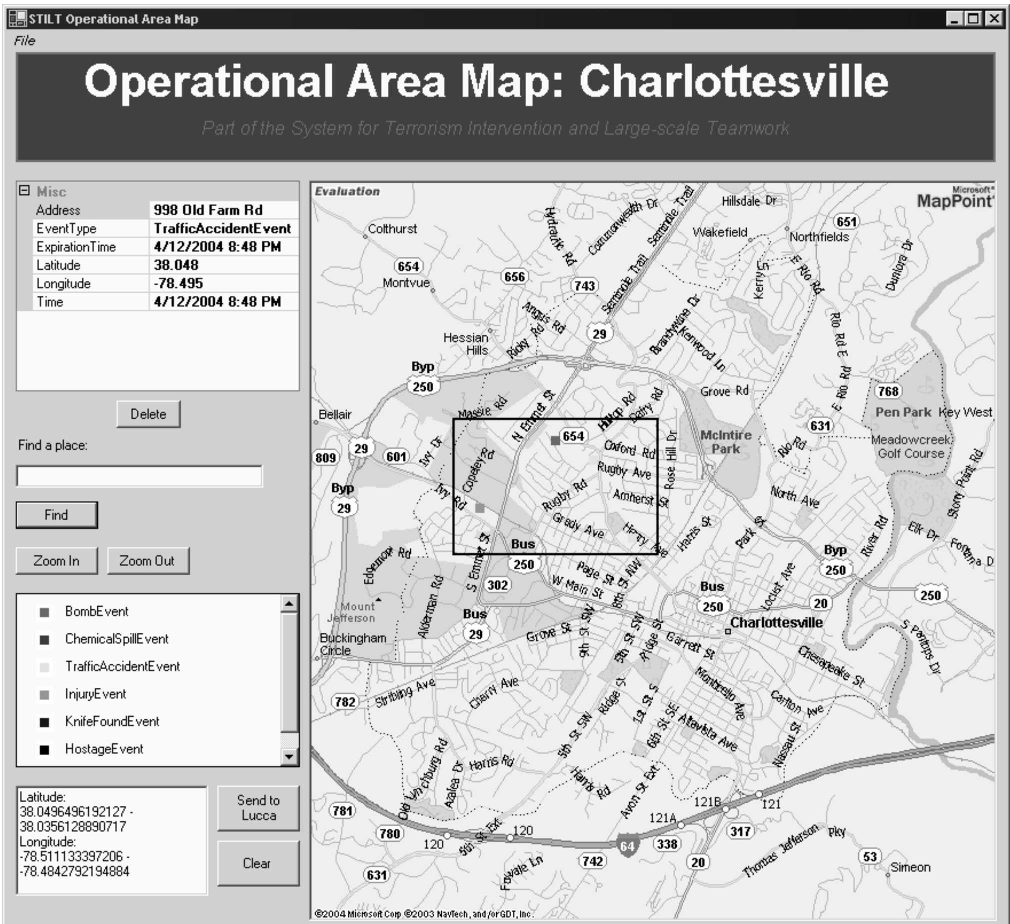


Figure 8: Operational Area Map Interface (with an area selected)

7 Conclusions

Recent terrorist attacks have demonstrated weaknesses in capabilities for detecting coordinated terrorist attacks and responding to them. To be effective, an emergency management system must perform three steps in a control loop. First, information must be distributed rapidly to those who will plan a response. Next, individuals must be directed to enact the response. Lastly, feedback must be collected, so that the response can be assessed and modified. This type of control loop is provided by the System for Terrorism Intervention and Large-scale Teamwork (STILT).

The purpose of STILT is to detect and respond to geographically-distributed, coordinated terrorist attacks. It can also be applied to general emergency response situations. STILT is a specific instantiation of the Willow architecture. It consists of Willow, formal specifications of the behavior of Willow, and various user interfaces for human interaction. The interfaces developed for STILT include Wiglaf, Lucca, and the

Operational Area Map. Wiglaf is a mobile command reception and event injection interface for personnel in the field. Lucca and the Operational Area Map are command center interfaces that allow decision makers to survey the state of affairs and inject workflows in response.

These interfaces make up the first generation of STILT software for human interaction. We are currently in the process of getting feedback from individuals in the fields of emergency response and emergency management. This feedback will be used to tailor the interfaces to better meet the needs of first responders and decision makers.

References

[1] Buck, George. *Preparing for Terrorism: An Emergency Services Guide*. Delmar, 1997.

[2] Dependability Research Group. “Willow Survivability Architecture.” 15 April 2004 <http://dependability.cs.virginia.edu/research/willow/>.

[3] Dwyer, Jim, Kevin Flynn, and Ford Fessenden. “Control: Discipline Broke Down in Eagerness to Help.” Firehouse.com. 15 April 2004 <http://www.firehouse.com/news/2002/7/7_P911d.html>.

[4] Hill, Jonathan, John Knight, Philip Varner, Dennis Heimbigner, and Alexander Wolf. “On the Implementation of Reactive Survivability In Critical Information Networks.” Unpublished paper. The University of Virginia, 2004.

[5] Knight, John C., Dennis Heimbigner, Alexander Wolf, Antonio Carzaniga, Jonathan Hill, Premkumar Devanbu and Michael Gertz. “The Willow Survivability Architecture.” Submitted to the International Conference on Dependable Systems and Networks (DSN 2002). Washington, DC, June 2002.

[6] Knight, John, Elisabeth A. Strunk and Kevin J. Sullivan. “Towards a Rigorous Definition of Information System Survivability.” Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX 2003). Washington, DC, April 2003.

[7] New York City Fire Department. “McKinsey Report - Increasing FDNY's Preparedness.” <http://www.nyc.gov/html/fdny/html/mck_report/toc.html>.

[8] Rowanhill, Jonathan C., Philip E. Varner and John C. Knight. . “Efficient Hierarchic Management For Reconfiguration of Networked Information Systems.” Accepted to the International Conference on Dependable Systems and Networks (DSN 2004). Florence, Italy, June 2004.

[9] Sullivan, K., J.C. Knight, X. Du and S. Geist. “Information Survivability Control Systems.” Proceedings of ICSE 21: Twenty-first International Conference on Software Engineering, Los Angeles, CA, May 1999.

[10] University of Colorado Software Engineering Research Laboratory. “Siena Publish/Subscribe Wide-Area Event Notification.” <http://www.cs.colorado.edu/~carzanig/siena/index.html>.

[11] Varner, Philip E. and John C. Knight. “An Information System for Emergency Response.” Unpublished paper. The University of Virginia, 2004.