

# Toward Implementation of Distributed Alerts in a Mobile Disaster Response Information Support Platform

Alan Calvitti, PhD<sup>1</sup>, Stephen W. Brown, MS<sup>2</sup>, Ricky Huang BS<sup>2</sup>, Fang Liu BS<sup>2</sup>,  
Leslie A. Lenert, MD MS<sup>1</sup>

<sup>1</sup>School of Medicine, University of California, La Jolla, CA

<sup>2</sup>Dept. of Computer Science and Engineering, University of California San Diego

*Multi-agency response to mass-casualty incidents is a complex socio-technical domain with unique communication and information management needs to provide real-time situation awareness. We describe technical and human factors challenges to implementation of distributed alerts in a mobile, wirelessly enabled informatics platform based on a flexible, non-locking remote object architecture and publish-subscribe messaging. A conceptual model organizes the "lifespan" of an alert into sequential stages from instantiation by decision logic, dissemination to appropriate responders, and presentation on client device interfaces. Use-cases considered for implementation include alerts based on vital signs telemetry, geo-location, and workflow management.*

## INTRODUCTION AND BACKGROUND

An alert system is a real-time decision support tool designed to signal a warning or relay a message triggered by specific patterns in input data, and present this signal to human operators for intervention or decision-making.

Mobile, transaction-oriented IT platforms such as the WIISARD system (see [wiisard.org](http://wiisard.org)) being developed to provide information support in disaster response settings in principle furnish the building blocks to support implementation of general alert system architectures. This flexibility makes it feasible to conceptualize a range of alert applications for medical and related work in disaster response.

In clinical settings, it's well known that technical and human factors limit the effectiveness of alert systems. These difficulties are also likely to be encountered in disaster response. In addition, the latter faces unpredictable, hostile environments, unreliable communication infrastructure and greater uncertainty in workflow and coordination. Design and implementation of effective alert systems in this domain is consequently a challenging task.

In this paper we begin a systematic description of these issues, and illustrate these concepts via use-case scenarios being considered for implementation in WIISARD. We introduce a

conceptual model of distributed alerts with sequential instantiation, distribution, and presentation stages. This model is dynamic and discrete-event oriented. The main goal is to help guide system development.

## Alerting Systems in Clinical Settings

In clinical settings, we encounter two distinct types of alert systems that fit the description in the introduction: alarms associated with physiological monitors and Computerized Clinical Reminders (CCRs).

Of relevance here is that physiological monitor alarms can be collocated with the monitoring devices, but in intensive care units they are often routed to watch stations instrumented for remote observation. The latter are examples of distributed alerts, although the routing is fixed in this case.

CCRs notify designated providers of the availability of patient-specific laboratory results, present them with a list of recommended clinical actions (e.g. prompting scheduling of preventive services) or alert them that other members of a patient's care team have blocked previously entered orders. CCRs are software components of clinical information systems e.g. VHA's VistA/CPRS. In this client-server system, reminders are presented to providers at the time when they access specific patient records. CCRs are also examples of distributed alerts because they are accessible from any client computer on a clinical network.

Experimental CCRs such as described by Reddy and co-investigators [1] leverage wireless communication networks to route alerts to alphanumeric pagers worn by providers in an attempt to decrease the latency between the time a reminder is generated by the information system to the time it is presented to the provider (particularly in outpatient care contexts, this latency can be substantial.) Delivery of such wireless reminders can be assumed a reliable operation in clinical settings. In contrast, the currently developed data networking protocols and infrastructure that would be deployed ad-hoc at the scene of a disaster are less robust.

## ISSUES FOR ALERT SYSTEM DESIGN

### **Components and Stages of Alert Systems**

The technological components shared by alerting systems include: 1) autonomous sensors or databases as sources of data. Human operators may also originate or modulate this input data. 2) A decision logic module where input data are aggregated. This module, typically a “headless” software agent, embodies a predicate that triggers onset of an alert by comparing observed input data against a model of actionable patterns. We call this the *instantiation* stage. Subsequently, this module can switch off or update the alert based on new observed data. 3) A user interface to display the alert signal or message. This is the *presentation* stage. As described earlier, these stages need not be embodied in one machine but may be distributed and linked by a communication system and we refer to the routing or delivery scheme as the *dissemination* stage. An alert may be intended for a specific operator or simultaneously disseminated and presented to multiple operators. This stage-based conceptual model helps to decouple orthogonal design issues and emphasizes the real-time, discrete-event nature of alert systems, including concomitant latencies through each stage.

### **Information Content of Alerts**

The information carried by an alert may be binary, graded, or complex. In the simplest physiological monitors equipped with auditory alarms, the message is binary: the presence or absence of the audible warning. In contrast, the Common Alerting Protocol (see [oasis.org](http://oasis.org)) defines 5-level gradations for the urgency, importance, and uncertainty of an alert (plus additional message structure.) Graded alerts enable context-sensitive filtering, e.g. alerts less urgent than a prescribed threshold can be automatically suppressed without altering the design of the instantiation or dissemination stages. CCR’s are complex: providers must take into account not just the presence of a reminder but also interpret the clinical and workflow significance of its content.

As information density and complexity increases, it may be warranted to transition to graphical information displays at the presentation stage [2]. Although such dashboard-like displays strain the definition of ‘alert’, in terms of the model described here they can be thought of as alert systems that shift the burden of complexity from the instantiation to the presentation logic. Such systems require higher bandwidth at the dissemination stage. Trivial instantiation logic implies data is transmitted for presentation in a less “compressed” form.

### **Alert Effectiveness in Socio-Technical Systems**

Since the instantiation, dissemination, and presentation stages are arranged serially, intuitively the overall effectiveness of an alert system is the “product” of the effectiveness of each stage.

Measures of discrimination accuracy familiar to informatics researchers are only applicable to the instantiation stage. What factors inform effective design of the dissemination and presentation stages?

Routing a true-positive alert to an inappropriate operator within an organization is likely to result in the alert being treated as a nuisance and ignored, while an operator who is prepared to respond to it will not receive it. This inefficiency suggests the importance of modeling roles and responsibilities within an organization to guide design of the dissemination stage. To the degree that these roles and responsibilities are dynamic, adaptable routing capabilities should be part of the architecture. Modeling the communication patterns within an organization may also indicate the acceptable latencies.

Improving the effectiveness of the presentation stage is a subject of human-computer interface and cognitive science research. Studies of work under high cognitive load in various domains have called attention to human perceptual (e.g. change blindness, habituation) and cognitive (e.g. single focus of attention) limitations, and even social factors (e.g. group pressure to ignore alerts) that reduce the effectiveness of otherwise carefully designed alert systems [3]. A true-positive alert disseminated to appropriate operators but masked by environmental noise or by other alerts (“alarm cacophony”) is in effect a false negative.

More generally, an alert indicating a fixed level of urgency may be perceived to aid decision-making in one workflow context yet be disruptive in another. Unfortunately, shifting workflow contexts are typically not observable to the decision logic at instantiation, as has been observed in the context of CCRs [4].

### **Situation Awareness in Mass-Casualty Response**

Multi-agency response to mass-casualty incidents is a complex socio-technical domain with unique information management needs.

Situation awareness during response is distributed. Although it is neither necessary nor desirable for all responders to possess global information, poor situation awareness poses safety risks, and hinders response coordination. This can be attributed to a variety of human and technical factors: communication systems deployed by multiple agencies and jurisdictions may not be interoperable and coordinating communication channels through

which alerts are often disseminated can be time-consuming; in rapidly changing situations, information recorded on paper forms or communicated verbally rapidly becomes stale; the command and control structure has been observed in drills and actual mass-casualty incidents to be adaptive, deviating from nominally hierarchical models. This contributes to confusion as to roles and responsibilities for specific response tasks. The challenge for alert system design is identifying the most appropriate responders for specific alerts types.

## WIISARD SYSTEM ARCHITECTURE

WIISARD is an experimental, 802.11-based wirelessly enabled mobile information support platform being developed to enhance medical and related aspects of response to mass-casualty incidents. WIISARD addresses the shortcomings of current tools, augmenting verbal, gestural, voice-radio and paper based communication and record keeping to facilitate triage, field-treatment, transport coordination and vital-sign telemetry of victims while providing increased situation awareness to responders.

### Component Hardware Devices

Medical responders can be categorized as first-tier (who directly care for victims), supervisors (who coordinate first-tier teams) and commanders. WIISARD applies the “system structure principle” and mirrors human organizational structure [5]. Thus first-tier responders carry compact wireless personal digital assistants (PDAs). Supervisors carry tablet-type computers, and medical commanders are stationed at command centers equipped with large display devices. Victims are instrumented with custom wireless triage tags and pulse oximeters (POXs). Each of these types of device runs client applications tailored to specific roles. Each device has specific user-interface constraints, e.g. PDAs have limited screen dimensions – every pixel is accounted for – whereas much more information can be simultaneously presented on command displays. Patient tags have custom interfaces to display triage status and alerts via color-coded LEDs. These devices communicate via an ad-hoc 802.11 network enabled by field-deployable wireless access points (APs).

In the current version of the WIISARD architecture, a single (centralized) server that interfaces to a database mediates data sharing. A custom middleware layer provides services that allow information to automatically and transparently propagate to all equipped responders.

A planned distributed version of WIISARD aims to replace the centralized server with multiple database servers to better support localized teamwork mitigating network faults. In this paper we only consider the current, centralized-server architecture. Key issues of alert system design expected in the next-generation WIISARD are also present in the current system. There is no inconsistency in using the term ‘distributed alert system’ in the centralized server context, as distributed refers to the potential for alert instantiation and presentation on distinct devices using existing WIISARD services for dissemination.

### Intermittent Connectivity

A key limitation of mobile IT platforms is that even in controlled environments it is unfeasible to maintain connectivity of roaming devices at all times. This has a spectrum of causes, e.g., physical-layer interference from buildings or even (large, metallic) fire engines, congestion-control timeouts, or simply roaming out of wireless coverage.

The validity of this intermittent connectivity assumption is underscored by the technical problems plaguing the recently tested \$140 million New York City Metro Transit Authority’s subway radio system designed to improve voice radio links between on-ground personnel and responders to emergencies underground. Despite pre-positioned amplifiers and hundreds of miles of antenna cable deployed in tunnels, NYPD and FDNY concluded that there were too many dead spots where radio signals did not reach or garbled communications [6].

WIISARD assumes intermittent connectivity of roaming devices from the start and contains features to mitigate its effects, increasing the effectiveness of time-critical group-oriented work under this constraint. These mechanisms are detailed in [5] and summarized next.

### Data Sharing and Transaction Control Model

The core of the WIISARD data-sharing and transaction control mechanism consists of shared cached object model that encapsulates underlying application data. These objects are distributed among client devices through a non-blocking publish-subscribe (pub-sub) messaging protocol. Consistency of the server-hosted persistent data is enforced through a conflict resolution scheme that uses precondition semantics only when conflicts are encountered, rather than locking data a priori.

Client applications that need to read or manipulate application data do so by subscribing to the object that encapsulates it. The non-blocking semantics and pub-sub enables a live, reactive system over intermittent or congested connections. Multiple

responders can contemporaneously read and modify the same object; what they actually receive and modify are copies cached by their respective clients. The cached object model opportunistically (i.e. when connected) tries to synchronize any modifications to the server's object mode state. In the current system, the server is the single "arbiter of truth".

### **Precondition Semantics and Ontological Modeling**

There is the possibility that distinct clients modify the same object. A conflict resolution protocol based on programmable precondition semantics determines which modifications are applied by the server. To date, only the simplest precondition has been implemented: the server accepts modification to a field only if client and server agree on the field's old value.

Development of more sophisticated precondition semantics offers the possibility to resolve some types of conflicts in a consistent manner. This functionality is contingent on development of ontological models of response workflow patterns. Such models may be informed by: 1) temporal relationships, i.e. order of precedence or metric temporal relationships among captured events; 2) spatial or spatio-temporal relationships as described in [7] that depend on geo-location histories, e.g. proximity of a responder to a victim when modifications were made; 3) relationships inherent in the organizational structure, e.g. dependencies in team interaction patterns or command chain authority.

## **USE-CASE SCENARIOS**

### **Telemetry Alerts**

WIISARD POX units can stream pulse rate and blood oxygen saturation data as well as internally generated alerts based on simple decision-logic. It may be desirable to augment these internal alerts with more sophisticated instantiation models, e.g. taking into account statistical patterns of vital signs over long-time windows. In addition, POXs are geo-locatable. Alerts based on aggregated spatial statistics may pave the way for automated syndromic surveillance in the event of chemical, biological, or radiological threats.

### **Geo-location Alerts**

WIISARD roaming devices are real-time geo-locatable via 802.11-assisted trilateration. As this is a physical-layer operation, clients need not maintain a connection to the server. It is desirable to monitor the patterns of movement of providers, victims and equipment. These patterns can be overlaid on a map, but depending on the scale of response, it may not be

feasible to monitor individual movement patterns. This motivates geo-location alerts. Specific scenarios include alerting when a geo-tagged victim or responder wanders into a hot zone, and alerting when a geo-tagged victim moves out of a designated containment area. Designated responders may manually define the hot zones or containment areas via a map interface.

Figure 1 (top) illustrates: 1) the consequences of intermittent connectivity for geo-location alerts; 2) the "think at the edges" principle, which states that decision-support approaches in distributed systems are more likely to be robust when pushed to the edges of the network. This would dictate, e.g. that alert instantiation be client- rather than server-instantiated.

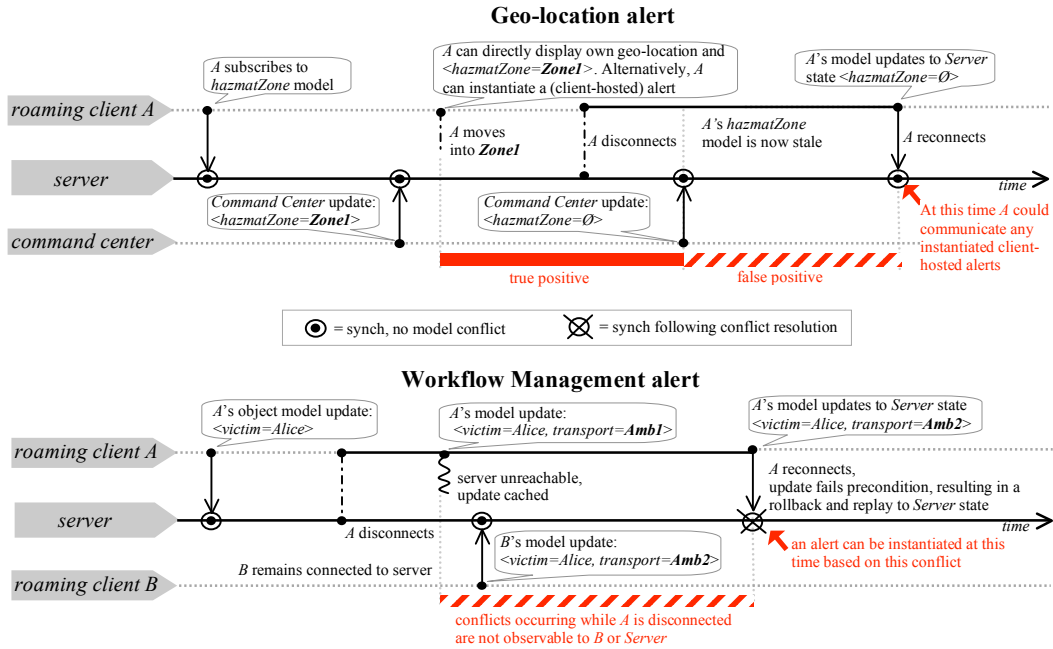
In the scenario illustrated here, a roaming client subscribes to hazmatZone objects. This client is automatically notified via pub-sub when hazmatZone objects are created, updated or deleted (provided the client is connected). The client performs its own geolocation and could directly display this relative to any zones, i.e. overlaid on a map. The assumption that hazmatZones will not be updated frequently allows the responder to continue to use this map for decision-making even while disconnected (provided it can reconnect periodically and frequently enough.) If the hazmatZone is deleted at the server end while client is disconnected, both a client-instantiated alert and the direct mapping method would result in a false positive (it is just as easy to construct examples yielding false negatives,) since the update will not reach the client. However, in the direct map visualization method, there seems to be more latitude to represent zone staleness [2].

### **Workflow Management Alerts**

Automated conflict resolution in the object model may not always be possible through the precondition mechanism- it may require expert knowledge and decision-making. In such cases it may still be possible to warn when potential or actual workflow errors or conflicts are identified. This motivates a class of alerts for workflow management.

The scenario depicted in Figure 1 (bottom), illustrates: 1) how intermittent connectivity may be the enabling cause of conflicts; 2) how these conflicts are resolved by WIISARD with the default precondition semantics; 3) an indication of the latency associated an alert identifying the occurrence of such a conflict (as opposed to potential conflict), based on the currently implemented default precondition semantics described earlier.

In the figure, multiple responders care for the same victim at various points in time. If at least



**Figure 1.** WIISARD’s non-locking distributed object and transaction model mitigate the effects of intermittent connectivity inherent in mobile IT. Here, idealized interaction of intermittent connections and object model transactions - using a trivial conflict resolution mechanism - is illustrated for plausible models of geo-location alert (**top**) and workflow management alert (**bottom**). In the latter, alerts may be used specifically to warn of such conflicts.

one responder disconnects, it is possible that they assign the victim to different ambulances unaware of the other’s actions (assuming such information is not collocated with victim). Upon reconnection, we see that the server accepts only one of the ambulance assignments (in this case the one occurred later in the sequence of events). The client whose assignment failed could log this conflict for post-event analysis or an alert may be instantiated. The server is also aware of this conflict, so in this case a server-hosted agent may also instantiate an alert for dissemination to, say, supervisors. The specificity of an alert warning of potential (as opposed to actually observed) conflicts of this nature would be too low to be of use, since it would be instantiated any time a client disconnects.

## DISCUSSION

The effectiveness of alerting systems in complex domains like disaster response will depend on the discrimination accuracy of the decision logic – an instantiation issue – but also on how the alert is subsequently disseminated to appropriate responders as well as its mode of presentation. Distributed lerts can be implemented on top of existing components and services provided by mobile, transaction-oriented information systems like WIISARD. Mobile IT is susceptible to intermittent connectivity that renders alert implementation challenging. WIISARD

employs a reactive object model with consistency maintained via programmable precondition semantics. Ontological modeling of response workflow could help inform development of these semantics to resolve or alert based on observed conflicts

This work is supported by NLM contract #N01-LM-3-3511

## REFERENCES

1. Reddy MC et al. Technology work and information flows: lessons from the implementation of a wireless alert pager system. *J. Biomed Informatics* 38:229-238, 2005.
2. Demchak B et al. Data quality and situational awareness during mass-casualty events. Submitted for publication, *AMIA National Symp.*, 2007.
3. Xiao Y et al. Organizational-historical analysis of the “failure to respond to alarm” problems. *IEEE Tr. Systems Man Cybernetics A* 34:772-778, 2004.
4. Saleem JJ et al. Exploring barriers and facilitators to the use of computerized clinical reminders. *JAMIA* 12:438-447, 2005.
5. Brown SW et al. Middleware for reliable mobile medical workflow support in disaster settings. *AMIA Annual Symp.*, 2006.
6. Radio system for subway gets a rebuff. *The New York Times* March 7 2007.
7. Guting RH, Schenider M. *Moving Object Databases*. Morgan Kauffman, 2005.