Siren: Context-aware Computing for Firefighting

Xiaodong Jiang[1], Kevin Wang[1], Nicholas Chen[1], Jason I. Hong[1], Leila A. Takayama[2], James A. Landay[1]

[1]Group for User Interface Research
Computer Science Division
University of California
Berkeley, CA 94720-1776, USA
{xdjiang, kwang, nchen, jasonh}@cs.berkeley.edu

[2]Department of Communication
Stanford University
Stanford, CA 94305-2050, USA
takayama@stanford.edu

[3]Intel Research Seattle
1100 NE 45th Street, 6th Floor
Seattle, WA 98105
james.a.landay@intel.com

Abstract

Based on an extensive field study of current firefighting practices, we have developed a system called Siren to support tacit communication between firefighters with multiple levels of redundancy in both communication and user alerts. Siren provides a foundation for gathering, integrating, and distributing contextual data, such as location and temperature. It also simplifies the development of firefighting applications using a peer-to-peer network of embedded devices through a uniform programming interface based on the information space abstraction. As a proof of concept, we have developed a prototype context-aware messaging application in the firefighting domain. We have evaluated this application with firefighters and they have found it to be useful for improving many aspects of their current work practices.

INTRODUCTION

Each year, fires kill about 4,000 civilians and 100 firefighters in United States alone [1]. Firefighting is a dangerous profession that calls for quick decisions in high-stress environments, constant reassessment of dynamic situations, and close co-ordination within teams. The smoke, heat and noise in a structure fire mask the environment and force firefighters to operate with an incomplete picture of the situation. "Firefighting is making a lot of decisions on little information," said one firefighter we interviewed. Improvements in information gathering, processing and integration can help firefighters work more effectively to prevent injury and loss of life, as well as minimize property damage.

The pervasive computing community itself can also benefit from research in this area. The nature of emergency response is fundamentally different from office environments, in terms of physical risk, psychological state, and operating conditions. This poses unique challenges for designers and researchers in terms of context awareness, new interaction techniques, and information visualization, to name a few. If we can make an impact in this highly stressful domain, where the systems we offer are secondary to the primary task, we might also be able to apply these results in less extreme environments for a wider audience, such as computing while driving.

From an extensive four-month field study of current firefighting practices, we found that firefighters often need to exchange information about their situation and their surrounding environment in a spontaneous and opportunistic manner. Such interaction is especially useful when firefighters need to be alerted about imminent dangers. This type of interaction needs to be *spontaneous* because the time when information exchange will occur depends on the dynamically changing situation and often has to be done without direct human initiation. It also needs to be *opportunistic* because the constant movement of firefighters in a complex urban structure makes it impossible to maintain an always-on communication channel among them.

The problem is that spontaneous and opportunistic interactions among firefighters are not well-served by current systems. Today, most firefighters rely on two communication channels on the scene of a fire. The first is a broadcast channel for voice communication. The second is a data broadcast channel for status update between incident commanders and dispatchers at a centralized emergency response center. Both channels use the 800MHz to 900MHz radio band. Often, only the voice channel or data channel can be used at a given time. Moreover, both channels are broadcast driven and manually operated to support explicit communication rather than the tacit communication needs between firefighters.

Advances in pervasive computing technologies provide us with an opportunity to let firefighters "see through the eyes of fellow firefighters" and provide a greater understanding of the overall situation. Small, cheap, wirelessly networked sensors (such as smart dust [2]) can be deployed on firefighters and in buildings, allowing us to gather contextual information—such as temperature, sound, movement, toxicity, and a person's location—at a level never seen before. The wealth of sensor data about firefighters and the environment can be exchanged between firefighters to help improve safety and effectiveness.

To support spontaneous and opportunistic interactions between firefighters, we have developed Siren, a peer-to-peer context-aware computing architecture that gathers, integrates, and distributes context data on fire scenes. To make tacit communication between firefighters more robust in the face of an inherently unreliable transport, Siren offers

multiple levels of redundancy in communication and feedback. Siren also simplifies development of emergency response applications by providing a uniform programming interface based on the information space abstraction [3]. Using Siren, we have developed a prototype context-aware messaging application and conducted an evaluation of this application with firefighters. They have found it useful for improving many aspects of their current work practices.

The rest of this paper is organized as follows. We discuss related work in section 2. An example search and rescue scenario is given in section 3 to illustrate how Siren-enabled applications may assist firefighters. In section 4, we describe key findings from our field study of current firefighting practices, and how they motivate the design of Siren. Section 5 describes key components of the Siren architecture, including a programming model based on the information space abstraction, storage management, communication, and the context rule engine. We describe a prototype context-aware messaging application developed using Siren in section 6, and our evaluation of it with firefighters in section 7. We discuss some lessons we have learned about designing for mission-critical pervasive computing applications in section 8. We conclude in section 9 and discuss future work.

RELATED WORK

There has also been a great deal of work at providing programming support for various aspects of pervasive context-aware computing. This includes the PARCTab system [4], Cooltown [5], the Context Toolkit [6], Contextors [7], Limbo [8], Sentient Computing [9], Stick-E notes [10], MUSE [11], SpeakEasy [12], Solar [13], XWeb [14], GAIA [15], one.world [16], and iRoom [17]. Most of them are designed to support office work in traditional work environments. Siren, on the other hand, aims to support field work practices of mobile firefighters.

Context Fabric [18] is a generalized service infrastructure for context-aware computing that implements an information space abstraction [3] and a P2P infrastructure. Siren implements Context Fabric for a peer-to-peer network of PDAs, and provides new capabilities for discovery, multi-hop communication and rule-based adaptation.

Other systems have also attempted to support emergency responders using mobile ad-hoc mesh networks (MANET). Draco [19] aims to develop rapidly deployable networking technologies that can support voice and video communications between emergency responders. In the commercial world, companies such as MeshNetworks are developing a self-forming, self-healing technology that automatically creates a wireless broadband network at an incident [20]. In Siren, we employ standard 802.11b networking technology and focus instead on supporting tacit communication needs between firefighters at the application level. While all MANET work focuses on design of the underlying communication protocol, our goal in Siren is to integrate sensing, communication and feedback in a single architecture to support tacit communication needs of firefighters.

The Command Post of the Future [21] is a set of projects investigating command in battlefield situations. The focus is on developing technologies for mobility and better decision-making, including multimodal interaction, information visualization, and knowledge-based reasoning. We complement this work by looking instead at tacit interactions between firefighters within a building, focusing on how the underlying software infrastructure can be designed to better support spontaneous and opportunistic communication.

Our work is also related to Camp et al, who looked at communication issues in emergencies and prototyped a radio system that would reduce voice congestion while maintaining situational awareness [22]. In contrast, we concentrate on supporting tacit data communication needs between firefighters.

SIREN-ENABLED SEARCH AND RESCUE: A SCENARIO

In this section, we describe a scenario motivated by our field studies to illustrate what tacit communications needs we intend to support.

Three firefighters conduct a search and rescue task in an office building. Each firefighter carries a PDA. Wireless-enabled sensors have either been placed at strategic locations (e.g. on smoke detectors) throughout the building, or have been deployed on the fly by initial fire response team whose duty is to size up the situation.

As firefighter F1 walks into the building, his PDA continuously monitors surrounding temperature from nearby sensors. F1 approaches the south exit of the building when his PDA detects an usually high heat level and alerts him of imminent danger. As a result, F1 avoids the south exit. At the same time, his PDA also detects the presence of firefighter F2, who is conducting a search and rescue task in an adjacent room, and automatically sends a message containing information about the south exit to F2's PDA.

A few minutes later, F2 receives an immediate evacuation order from the incident commander. F2 knows that there is a growing fire hazard around the south exit, so he rushes toward north. On his way to the north exit, F2's PDA detects the presence of firefighter F3, who is running toward the south exit, unaware of the potential danger. F2's PDA automatically forwards the message received from F1 to F3's PDA, which immediately alerts him of the danger. As a result, F3 turns back to run toward the north exit, safely leaving the building.

FIELD STUDY OF FIREFIGHTING PRACTICES

To inform the design of Siren, we conducted a field study that spanned four months and included over 30 hours of interviews and low-fi prototype evaluation with 14 firefighters in three fire departments. The firefighters were from many levels of the organizations, including one assistant chief, four battalion chiefs, two captains, two engineers and five firefighters.

We employed several investigation methods for our study, including interviews, training observations, and field observation. We conducted interviews at fire stations while the firefighters were on duty and learned about their organizational structure, tools, routines, regular interactions, and typical environment. By observing one field exercise at a training tower used to train new firefighters, we received a first hand sense of how firefighters tackle an urban structure fire. In addition, we accompanied firefighters on two emergency calls to see first hand how they accomplished their tasks.

A full account of our field study can be found in [23]. We have looked at many aspects of current emergency response practice. In this section we give an overview of some findings that help motivate the key design decisions in Siren.

Figure 1. We conducted a four-month field study with firefighters in their work environment. This figure illustrates one such environment: a mobile incident command located at the back of a command vehicle.

Key Findings

First, we found that **firefighters often have an incomplete picture of the situation**. Collecting information on scene is difficult because it currently involves sending firefighters into unfamiliar areas with many potential hazards. The dynamic nature of fires quickly reduces the certainty and validity of collected information. Experienced firefighters compensate for this lack of information by making quick visual assessments of the status of a fire. However, fires in walls, attics, roof lines, and other obscured areas are hard to detect. Hidden hazards such as a structurally weak floor or a potential backdraft may cause the floor to collapse or an oxygen-starved fire to suddenly explode. Effective detection of hidden hazards and prompt alerts are critical to avoiding fatalities.

Our second finding is that **communication needs on fire scenes are poorly served by current systems**. Communication is currently done face-to-face or through radio on pre-specified frequencies. However, our interviewees noted that noise intensity and radio congestion are serious problems. One firefighter we interviewed said, “There is a lot of noise on the fire ground. You’re inside; the fire is burning; it makes noise; there’s breaking glass; there’s chain saws above your head where they’re cutting a hole in the roof; there’s other rigs coming in with sirens blaring; lots of radio traffic; everybody trying to radio at the same time.” Radios operate as a broadcast channel where everyone can hear everyone else. Another firefighter explained, “I’m usually listening to at least three [radios]. It’s tough, and then you’ve got people calling on the cell phone at the same time.” There also exists the problem of radio dead zones. Radio signals often get blocked in certain parts of structures, such as basements or in high rises. Communications taking place in those areas often cannot get through. This problem caused serious problems for World Trade Center responders, where many firefighters missed the evacuation order [24, 25].

Third, **firefighters operate in extremely harsh environments during a fire**. A typical firefighter wears 40 to 60 lbs of equipment. As one firefighter put it, “Crawling down a hallway blackened with smoke, 1500 degrees, wearing fire gear, lens fogged over with steam, one hand up and one hand down – it isn’t as easy as you think.” Given such a harsh and changing environment, any system needs to support reasonably robust communication from an inherently unreliable underlying transport.

From the Field to Design

Findings from our field study have motivated key design decisions in Siren, including a peering architecture, a “store-and-forward” communication model, and multiple levels of redundancy in communication and user alerts. Supporting a resilient architecture

All emergency response systems have an underlying issue of reliability. They need to withstand extreme physical conditions as well as survive failures like radio dead zones and dead sensors. This suggests that context-aware systems should operate independently without having to rely on any external infrastructure, but also take advantage of additional support when possible. For example, a firefighter that is cut off from his companions should still have some base level

of context support from whatever sensors and old data he is carrying with him, and get better sensor data and processing if it is available. Architecturally, this suggests a peering model, where each firefighter carries an autonomous peer that can federate with others to increase the level and accuracy of shared information.

Supporting Tacit Interactions

Our field study reveals the need for supporting spontaneous and opportunistic interactions between firefighters on a fire scene. Such interactions are especially useful when firefighters need to be alerted about the potential danger to themselves, and to their fellow firefighters. This interaction model has the following characteristics:

1. Interactions are often triggered by external events (e.g. requests from fellow firefighters or changing environmental conditions), and not by conscious human efforts
2. Such interactions are usually brief (e.g. exchange of small amount of information about firefighters and the environment they operate in) and do not have the strict real time delivery constraint of voice or video communication
3. The times when such interactions take place are hard to predict. Many factors, such as proximity between firefighters and complex environmental impact on radio channel, will determine the exact moments such interaction has an opportunity to happen.

To support tacit interactions among firefighters, devices in a Siren network can discover each other when they are within 150 feet of each other and initiate message exchange. Moreover, a "store-and-forward" model is adopted by queuing all intermediate messages. Until the message expires, it avoids get lost simply due to the absence of network connectivity. All messages are time stamped and assigned an expiration period, which are checked periodically by all receiving devices. This helps ensure the freshness of data that some applications may demand.

Supporting multiple levels of redundancy

Through our field study, we found that redundancy, not typically found in consumer applications, becomes an important requirement for firefighting applications. In designing Siren, we aim to support multiple levels of redundancy, in both communication and user alerts.

Siren supports communication redundancy on multiple levels. Messages are exchanged between devices over multiple hops in a peer-to-peer network. As such, messages generated by one device may reach another device through multiple paths and a different set of intermediate nodes. This "path redundancy" maximizes the chance that an important message will eventually be received by a device that may need it.

The second redundancy feature is "storage redundancy". Because of the "store-and-forward" model, any message may reside on multiple devices at a given time. This in turn maximizes the opportunity for any other device to receive a message.

Another choice we have consciously made in Siren's communication design is "version redundancy". Multiple versions of information about the same firefighter captured at different time are stored in the message queue on a receiving device. This way, important information about the environment isn't lost.

Siren applications also support redundancy in user alerts. Instructions and warnings issued by firefighters or incident command will be sent over Siren's multi-hop data communication channel, in addition to being broadcast on the traditional voice channel. This helps address the problems of noisiness, congestion, and dead zones on traditional broadcast radio channels whose signals originate from outside of the building. Moreover, alert messages generated by Siren can also be displayed in different ways, and in different modalities.

SIREN ARCHITECTURE

Siren is a peer-to-peer context-aware computing architecture that supports integrated sensing, communication, and user alerts. Applications running on top of Siren gather information from sensors carried by firefighters or embedded throughout the environment, communicate with each other in a P2P fashion, and provide context-dependent feedback to firefighters. Siren is implemented in 5000 lines of C++ code and currently runs on WiFi-enabled Pocket PCs.

The Siren architecture consists of three interacting components: a storage manager that implements the information space abstraction; a multi-hop communication manager; and a context rule engine (see Fig. 2). All three components and applications running on top of Siren interact via the InfoSpace API, a tuplespace-like [26] programming interface that loosely couples the whole system.

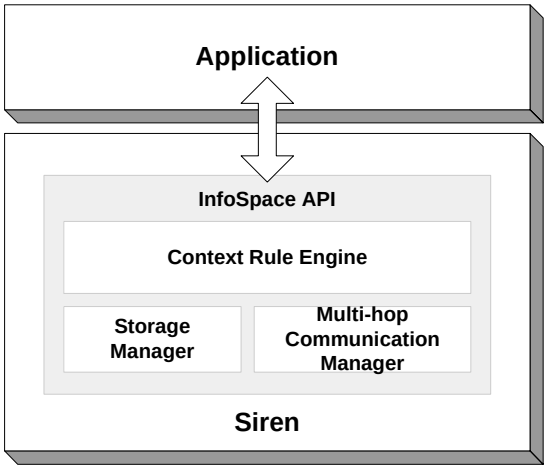

Fig. 2. Siren uses the InfoSpace API to unify storage, communication, and adaptive feedback via context-dependent rules.

The InfoSpace Programming Model

The key abstraction in Siren is the InfoSpace, a set of tuples associated with a single entity such as a firefighter, a room, or a device (Figure. 3). These tuples contain "tags" that describe the type, intended recipient, and any access control rules that must be enforced whenever tagged tuples are processed. For example, a firefighter's InfoSpace may include temperature data that is collected and stored on multiple devices. InfoSpace operators carry out processing operations on tuples in that InfoSpace, such as the insertion or retrieval of tuples.

Siren provides a unified programming model for developers of emergency response applications by treating sensing, storage, communication, and feedback as InfoSpace operators. These operators interact with each other through either local procedure calls or remote message passing if they belong to InfoSpaces residing on different devices. As such, the InfoSpace metaphor provides a decentralized way of combining diverse functionality into a unified, feature-rich system.

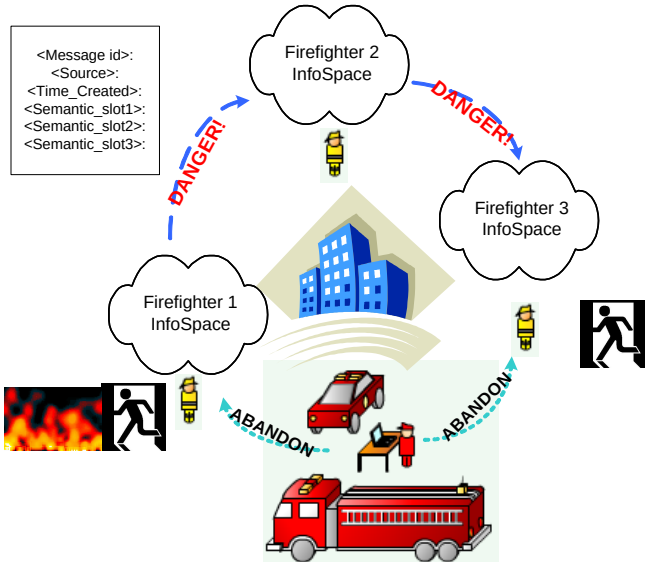

Fig. 3. Siren InfoSpaces interact through operators

The Storage Manager

The dynamic nature of firefighting requires loosely-coupled systems that are easily scalable. Siren's storage manager is an adapted implementation of the Context Fabric [18] on a peer-to-peer network of embedded devices and meets the scalability needs using an XML-based communication protocol. This component provides the API used to manipulate InfoSpaces, and manages the storage, querying, and retrieval of tuples.

The InfoSpace API consists of four operations: inserting a tuple, retrieving a tuple, subscribing to an InfoSpace, and unsubscribing from an InfoSpace. InfoSpaces are described using an HTTP-like URL with the associated device's name, communication port, and InfoSpace name. Tuples can be retrieved via a unique handle or by an XPath query [27]. For example, an application may retrieve all tuples that are tagged as high in temperature with the query, "`//Temp[@value="high"]`". Any operator, application or component in Siren, can "subscribe" to an InfoSpace to request to be notified when new tuples arrive and can also specify certain types of tuples of interest with a long-standing query.

Siren's InfoSpaces are used to support both storage and communication through two types of tuples: permanently stored tuples and temporary tuples for communication purpose. By default, all tuples are permanently stored upon insert, but their tags can specify that they will only be stored for a limited period of time.

Each firefighter owns an InfoSpace that contains data about his location, surrounding temperature, and the level of remaining oxygen supply. All this data is gathered in real time from sensors embedded in the environment or attached to firefighters. Each device in a Siren network stores both an InfoSpace of its own and a "snapshot" of InfoSpaces owned by other devices.

An InfoSpace snapshot is a point-in-time copy of the aggregation of data contained in an InfoSpace during a given time window. It is a view maintained by each device of other peers in a Siren network. Suppose an InfoSpace contain a firefighter's location, surrounding temperature and remaining oxygen supply. Each piece of data is gathered from sensors on a per-second basis. To obtain the snapshot of the InfoSpace for a time window of 10 seconds, each device will check data it has received about that InfoSpace's status during the past 10 seconds, and aggregate them. During aggregation, average will be computed for numeric sensor readings (such as temperature and remaining oxygen supply) while majority voting will be used to determine the aggregation of ordinate sensor readings (such as location).

Snapshots are done independently by each device in the network at a predefined time interval. It represents a device's "view" of the status of other devices at a given time. Because of the unreliable nature of the radio network in an urban structure, not all messages will be received by all devices during a given time window. Thus snapshots of the same InfoSpace may be different for different receiving device at any given time. We do not attempt to maintain consistency between snapshots taken by different devices because a weakly consistent system will improve overall availability and be more resilient to network partitions.

The Multi-hop Communication Manager

When Siren receives a request to insert to or retrieve from a remote InfoSpace, the sending device delegates the request to the appropriate remote host by formatting the request into an XML message and sending it to the peer over HTTP. A receiving device parses the incoming XML message and handles the request in a way similar to a local request, returning the result over HTTP. Siren operators can broadcast select tuples (such as "help" messages) to its known hosts, which can then be rebroadcast at the remote device, effectively broadcasting tuples over multiple hops of peers.

Siren supports discovery and time-based message queuing. Each firefighter's device discovers all neighbors currently in range, and broadcasts to all neighbors a message containing the current InfoSpace owned by the device. Each message is structured into the following format:

```
Message id: ….
Source: ….
Time_Created:
Semantic Slot 1:
Semantic Slot 2:
Semantic Slot 3:
```

Each semantic slot represents a different type of sensor reading, such as location or temperature, and contains the following fields:

```
Type: location
Value: 525 Soda Hall
Confidence: 70%
Expiration:
```

The "expiration" field indicates the time after which information contained in this slot becomes invalid. When receiving a new message, a device first checks the "message id" to avoid using a message twice in producing snapshot of an InfoSpace. On the other hand, Siren will continue forwarding a message it has seen before until all semantic slots contained in the message expire. Although this may add some redundant messages to the network, we believe such redundancy is useful to achieve the maximum reliability of message delivery.

The device will also check the expiration time of each semantic slot in the current message. If information contained in a semantic slot expires, it will invalidate the slot by rewriting "null" into it before forwarding the message. If all semantic slots in a message expire, the device will stop forwarding the message to others. This way we avoid network flooding by ensuring that no message will travel around the network forever.

Each Siren device maintains its own message queue. Each message queue contains all received messages that originate from the same InfoSpace (device). Messages in each queue are sorted according to the time they are created. A message is marked "sent" when it is forwarded to all neighbors. It will be permanently removed from the queue when a snapshot is taken using this message.

Each message is uniquely identified by its "message_id" and "source" of creation. There is a subtle point here about why we choose not to overwrite older messages as newer versions arrive based on these two attributes. This decision is made based on special requirements of the firefighting domain we want to support. For example, a firefighter may be in room A when a message was generated indicating a high surrounding temperature. A few moments later he may move to another room B with a normal surrounding temperature. Even though sensor readings in the old message no longer accurately reflect the current status of the firefighter, the knowledge that room A has high temperature and therefore may pose danger to other firefighters in the future is nonetheless very useful. This information will be lost if we choose to overwrite all old messages with new ones. This is another example of how communication redundancy is supported in this environment.

Message queuing provides an extended opportunity for interaction between firefighters. Firefighters who are not immediately adjacent to each other can exchange messages at a later time provided that the message being exchanged is still valid.

The Context Rule Engine

The "brain" of Siren is a context rule engine that takes InfoSpace snapshots as input and interprets them for generating user alerts. The rule engine is a production system that processes application-defined context rules, resolves potential conflicts and outputs reformatted "alert messages" back to the application.

The context rules are structured in the canonical "if…then…" format. The "if" part specifies conditions based on aggregated sensor readings contained in snapshots. The "then" part specifies a "alert message" that is to be output by the rule engine when the condition is true. Alert messages are used by Siren applications to render user feedback in appropriate modalities. Siren currently supports five types of alert messages: (1) "a dangerous place" (2) "danger to oneself" (3) "next to a dangerous place" (4) "others in danger" (5) "instructions".

"A dangerous place" indicates that a specific location in the urban structure poses potential danger to firefighter safety in the immediate future, such as an unusually high

surrounding temperature. “Danger to oneself” signals imminent danger to safety of a firefighter himself, such as a low level of oxygen supply. “Next to a dangerous place” signals that the firefighter is adjacent to a location that is considered a “dangerous place”. “Others in danger” signals that there are other firefighters nearby who face “danger to oneself”. “Instructions” signal orders or requests from incident command or fellow firefighters such as “abandon” or “help”. It should be evident that multiple rules may fire on the same snapshot. Consequently multiple “alert messages” may be produced.

Application of many context rules in Siren depends on access to a location model to decide on spatial notions such as adjacency. In their current practices, firefighters divide different locations within an urban structure along the direction it is facing and name them sides A to D. We have similarly divided each floor into multiple sections, lableled from A to D, and maintain their neighboring information

Each Siren application defines its own set of context rules and store them in a configuration file. When an application starts, it supplies the name of configuration file to Siren’s rule engine and rules contained in the file are subsequently applied. Currently, this process is done once at the start of an application.

A PROTOTYPE CONTEXT-AWARE MESSAGING APPLICATION FOR FIREFIGHTERS

Using Siren, we have developed a prototype context-aware messaging application to support the firefighting scenario we outlined in section 3. Each firefighter will carry a PDA on which the messaging application is running. Each PDA is attached with a Berkeley smart dust mote [28] that is used as both a local sensing device and a communication facility to gather data from other motes embedded in the environment. All PDAs are Wi-Fi enabled and can connect to each other in peer-to-peer mode using the underlying 802.11b protocol.

Our context-aware messaging application consists of two main components: a set of context rules and a user interface for feedback. One example context rule that’s included in the rule set is:

```
IF (firefighter F1 IN room A) AND
   (surrounding temperature > 1500F)
THEN (generate_alert(firefighter F1 in
danger)) AND
    (generate_alert(room A is a dangerous
place))
```

Through our field study, we have found that the two key requirements in designing user feedback in the firefighting domain are minimizing distraction and supporting redundant forms of output. Given the harsh environment firefighters operate in, they cannot be expected to perform sophisticated direct manipulation tasks on a complex interface. Moreover, traditional hands-free interaction techniques such as speech input is hampered by the noisy environment, and the fact that speech input still implies a model in which firefighters are required to attend to a personal device and manually retrieve information.

In our design, we have adopted an alert-driven interaction model in which firefighters need not interact with device except when he is to issue an instruction to fellow firefighters. All warnings and alerts will be delivered to the firefighter on an as needed basis. In designing the user interface, we avoid all interaction metaphors that would require direct manipulation, such as zooming, selection, scrolling, and text entry.

The visual display of the interface is designed to be as simple as possible to minimize visual distraction to firefighters (see Figure 4). We use simple shapes to represent firefighters (squares represent a firefighter herself, and circles represent other firefighters), and use color to represent their status. Previous research [29] has shown that mobile devices with micro-level form factors can be designed to convey critical information and provide effective notifications. Siren applications are designed for expert users who undergo extensive training in firefighting procedure and equipment. We believe simple design will be most effective for them.

As Figure 4 illustrates, we divide the PDA screen into three sections. The upper part of the screen overlays icons representing firefighters on top of a floor plan. The floor plan is divided into different locations according to the location model defined in section 5.3. Each location will turn red when it is considered to be a “dangerous place”. Different parts of the floor map will be shown depending on the current location identified through RF beams emitted by location-tagged motes embedded in the environment. This map section of the screen refreshes every 10 seconds to show the current location of firefighters. Each firefighter is represented by a circle labeled by his badge ID. The color of the circle indicates whether the firefighter is considered in danger: flashing red for imminent danger, red for “next to a dangerous place”.

The bottom left part of the screen is used to display alert messages generated by the context rule engine. There are three types of alert messages: “danger to oneself” (colored flashing red), “next to a dangerous place” (colored red) or “other people in danger” (colored yellow). The screen is divided into multiple panes, where each pane contains a new incoming alert message. Each pane will display the name of the firefighter, his location and the source of potential danger. Every 10 seconds a new alert message is displayed and an old message that has been on the screen for the longest time is discarded.

We use a priority queue for storing all alert messages generated by the context rule engine. Each incoming

message is automatically assigned a priority rating depending on its type: "danger to oneself" is assigned a priority rating of 5, "next to a dangerous place" is assigned 3, while "other people in danger" is assigned 2. Every 10 seconds, the priority ratings of all messages increase by 1. This mechanism ensures that eventually all messages will be displayed, regardless of its original priority assignment.

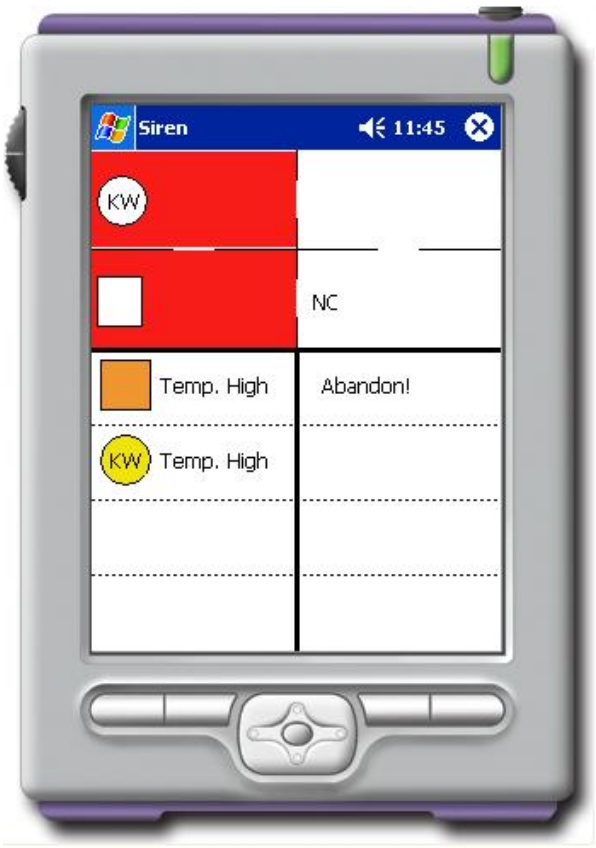

Figure 4. User Interface for Siren-based Context-aware Messaging Application. The upper part of the interface is a floor plan with four adjacent rooms. The lower left part contains user alerts generated by Siren, and the lower right part contains messages manually generated by fellow firefighters or incident commanders.

Information displayed on the bottom left pane is partly redundant with what's displayed over the floor plan, although in more detail. This design is consistent with our finding that firefighters prefer to see the high-level picture first, and details should be displayed only when needed. In this case, information about individual firefighters is only presented when needed. For example, detailed information about individual personnel is displayed in flashing red if a potentially critical danger is detected, such as high heat level.

The bottom right pane is used to display manual instructions and warnings issued by firefighters or the incident command, such as "abandon" or "help", as well as identity of the issuer. We have programmed the large physical buttons on Pocket PCs so that each button corresponds to a particular instruction or warning message. Firefighters can issue instructions or warnings by pressing these physical buttons. This frees them from having to perform interaction tasks on a small touch screen with a stylus. While firefighters can also issue these instructions over the traditional radio voice channel, this becomes a redundant form of output to maximize the chance of such instructions being received.

EVALUATION OF CONTEXT-AWARE MESSAGING APPLICATION WITH FIREFIGHTERS

We conducted a formative evaluation of our context-aware messaging application with six firefighters at a large Fire Department. The firefighters inspected the application, critiqued its functionality and design and brainstormed with us about its potential use. Overall firefighters found the application to be quite useful for enhancing many aspects of their current practices, including improved communication reliability, better firefighter safety, and a better size-up of dynamically changing situations.

Some interviewees commented that our application would be useful for providing adequate redundancy in communication, especially for large and complex urban structure such as hospitals, research labs, high rises and large warehouses. One firefighter commented, "I've been to many hospitals… there are lots of shielded walls and basements for surgeries, my 800MHz radio often doesn't work… The delayed hopping model you have will certainly help here."

Other firefighters would like to leverage the location tracking capabilities of our application to enhance firefighter safety. One firefighter proposed a new way of using our application, "imagine sending one firefighter whose job is to drop your sensors on each floor: in the room, on door wedges and apartment packs [apartment packs are hoses firefighters use to connect to water supplies]… when a firefighter is down, we will be able to know roughly where he is using your system… that is huge for us."

Our interviewees also envisioned our application be used in conjunction with other tools such as a thermal imager to narrow down the area to look in a typical search and rescue task. One firefighter commented, "This way you don't have to comb through the entire area and can avoid potential fire hazard, since your system will tell you which room is likely to be more dangerous."

Firefighters generally liked the simple UI design and the fact that no direct manipulation would be required of them. One battalion chief commented, "This is great…. I am not a techno-wizard….The level of detail here is just about right for me." Some firefighters also suggested that we implement a "multi-dimensional" view of the building and

make it easy to switch between views of rooms on different floors.

There are two concerns that our interviewees have expressed regarding the deployment of such application. First is the ruggedness of such technology, including both hardware and software reliability. Second, firefighters cautioned us not to rely solely on sensors embedded in the environment. “The most effective tools for us are always those we can directly access and maintain”, said one firefighter. Our interviewees suggested we could also attach smart dust sensors to firefighters’ breathing apparatus, because it is carried by all firefighters and is also uniquely assigned for each firefighter.

LESSONS FOR DESIGNING MISSION-CRITICAL PERVASIVE COMPUITNG APPLICATIONS

Through designing the concept context-aware messaging applications for firefighters, we have learned a few key lessons about the unique design requirements for mission-critical context-aware applications.

For users of mission-critical context-aware applications, interacting with a context-aware device is often not their primary focus. Their ability to perform sophisticated interaction tasks is also hampered by the harsh environment they operate in. Therefore, mission-critical context-aware applications should be designed to support implicit and opportunistic interactions. In our concept context-aware messaging application for firefighters, we have structured the communication design around the notion of “opportunity for interaction”. User alerts are given dynamically based on current context, and the message queue allows opportunities for interaction to be extended. Also, the interaction paradigm is centered on notification and alert, instead of requiring direct user input.

Tasks in mission-critical context-aware applications are often ill-defined. Because of the volatile environmental conditions, the actions through which goals can be achieved vary significantly. Through our experience with designing for firefighters, we have found that it is more useful to understand the complex work practice in terms of activities: actors direct actions at tools and artifacts. By analyzing the emerging patterns of activities and relationships between actors and artifacts, we can design our context-aware applications to support existing work practice.

Redundancy is often a less desirable feature for consumer-oriented context-aware applications because it increases complexity and degrades speed of interaction. However, in mission-critical context-aware applications such a feature is almost essential. In our concept context-aware messaging application design, we have employed redundancy in both feedback design and communication design. Redundancy extends the opportunity for interactions and therefore makes the user interaction experience more predictable.

In mission-critical context-aware applications, interaction between users and the context-aware computing system is not through an isolated device. The ability to into the physical environment in which users operate is critical for the success of such an application. In our concept context-aware messaging application design for firefighters, the point of interaction includes stairways, fire hoses and the rear of a command vehicle. A successful user interface design needs to take these diverse points of interaction into account.

CONCLUSION AND FUTURE WORK

Based on an extensive field study of current firefighting practices, we have developed a peer-to-peer architecture called Siren that provides the foundation for the gathering, integration, and distribution of context data on fire scenes.

It allows the development of emergency response applications on a peer-to-peer network of embedded devices through a uniform programming interface based on the information space abstraction. Siren is designed to support spontaneous and opportunistic interaction between firefighters and provides multiple levels of redundancy to suit the mission-critical nature of firefighting. Using Siren, we have developed a prototype context-aware messaging application to support a search and rescue scenario obtained from our field study. We have evaluated this application with firefighters and they have found it to be useful for improving many aspects of their current work practices.

We are currently investigating supporting alternative sensory modalities such as tactile output in Siren. We plan to develop more sophisticated Siren applications and user test them with firefighters in their training exercises.

ACKNOWLEDGMENTS

We thank the Alameda, Berkeley, and El Cerrito fire departments. We also thank Larry Leung for his contribution. This research was supported by NSF IIS-0205644 and CITRIS.

REFERENCES

1. U.S. Fire Administration, F.E.M.A., Facts on Fire. 2000. http://www.usfa.fema.gov/dhtml/public/facts.cfm
2. Estrin, D., Culler, D., Pister, K., and Sukhatme, G.: Connecting the Physical World with Pervasive Networks. IEEE Pervasive Computing 1 (2002) 59-69
3. Jiang, X. and Landay, J. Modeling Privacy Control in Context-aware Systems Using Decentralized Information Spaces. IEEE Pervasive Computing 1(3) (2002)
4. Schilit, B.N., *A Context-Aware System Architecture for Mobile Distributed Computing*, Unpublished PhD, Columbia University, 1995. http://seattleweb.intel-research.net/people/schilit/schilit-thesis.pdf
5. Kindberg, T. and J. Barton, A Web-based Nomadic Computing System. *Computer Networks* 2001. **35**: p. 443-456.
6. Dey, A.K., Salber, D. Abowd, G.D. A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications, Human-Computer Interaction, Vol. 16(2-4), 2001, pp. 97-166.

7. Crowley, J.L., J. Coutaz, G. Rey, and P. Reignier. Perceptual Components for Context Aware Computing. In Proceedings of *Ubicomp 2002*. Göteborg, Sweden. pp. 117-134 2002.
8. Davies, N., S.P. Wade, A. Friday, and G.S. Blair. Limbo: A tuple space based platform for adaptive mobile applications. In Proceedings of *The International Conference on Open Distributed processing / Distributed Platforms (ICODP/ICDP '97)*. pp. 291-302 1997.
9. Addlesee, M., R. Curwen, S.H. Newman, P. Steggles, A. Ward, and A. Hopper, Implementing a Sentient Computing System. *IEEE Computer* 2001. **34**(8): p. 50-56.
10. Pascoe, J. The Stick-e Note Architecture: Extending the Interface Beyond the User. In Proceedings of *International Conference on Intelligent User Interfaces*. pp. 261-264 1997.
11. Castro, P. and R. Muntz, Managing Context for Smart Spaces. *IEEE Personal Communications* 2000. **5**(5).
12. Edwards, W.K., M.W. Newman, J.Z. Sedivy, T.F. Smith, and S. Izadi. Challenge: Recombinant Computing and the Speakeasy Approach. In Proceedings of *Eighth ACM International Conference on Mobile Computing and Networking (MobiCom 2002)* 2002
13. Chen, G. and D. Kotz. Context Aggregation and Dissemination in Ubiquitous Computing Systems. In Proceedings of *Fourth IEEE Workshop on Mobile Computing Systems and Applications*. pp. 105-114 2002
14. Olsen, D.R., S. Jefferies, T. Nielsen, W. Moyes, and P. Frederickson, Cross-modal Interaction using XWeb. *CHI Letters, The 13th Annual ACM Symposium on User Interface Software and Technology: UIST 2000* 2000. **2**(2)
15. Román, M., C.K. Hess, R. Cerqueira, A. Ranganathan, R.H. Campbell, and K. Nahrstedt, Gaia: A Middleware Infrastructure to Enable Active Spaces. *IEEE Pervasive Computing* 2002. **1**(4): p. 74-83
16. Grimm, R., J. Davis, E. Lemar, A. Macbeth, S. Swanson, T. Anderson, B. Bershad, G. Borriello, S. Gribble, and D. Wetherall, *Programming for pervasive computing environments*. Technical Report UW-CSE-01-06-01, University of Washington Department of Computer Science and Engineering, Seattle, WA 2001
17. Johanson, B., A. Fox, and T. Winograd, The Interactive Workspaces Project: Experiences with Ubiquitous Computing Rooms. *IEEE Pervasive Computing* 2002. **1**(2): p. 67-74
18. Hong, J.I. and Landay, J.A.: An Infrastructure Approach to Context-Aware Computing. Human-Computer Interaction Vol. 16. (2001) 287-303
19. Agrawala, A. Draco: Connectivity Beyond Networks (2001) http://mindlab.umd.edu/research_draco.html
20. Mesh Netowrks, http://www.meshnetworks.com/
21. DARPA Information Exploitation Office, Command Post of the Future. http://dtsn.darpa.mil/ixo/programdetail.asp?progid=18
22. Camp, P.J., *et al.* Supporting Communication and Collaboration Practices in Safety-Critical Situations. In Proceedings of *Human Factors in Computing Systems: CHI 2000*. Fort Lauderdale, FL: ACM Press. pp. 249-250, 2000.
23. Jiang X., Hong J., Takayama, L. Ubiquitous Computing for Firefighting: Field Studies and Large Displayes for Incident Command. To appear in CHI 2004. Vienna, Austria
24. McKinsey and Co, Increasing FDNY's Preparedness (2002) http://www.nyc.gov/html/fdny/html/mck_report/toc.html
25. Paulison, R.D.: Working for a Fire Safe America: The United States Fire Administration Challenge (2002) http://www.usfa.fema.gov/dhtml/inside-usfa/about.cfm
26. Rowstron, A. Using asynchronous tuple space access primitives (BONITA primitives) for process coordination. In: Garlan, D. and Le Métayer, D. (eds.): Coordination Languages and Models. Lecture Notes in Computer Science. Springer-Verlag , Berlin (1997) 426-429
27. XML Path Language (XPath), W3C, http://www.w3.org/TR/xpath
28. Kahn, J.M., Katz, R.H., and Pister, K.S.J.. Mobile Networking for Smart Dust. Proceedings of MobiCom 1999: The Fifth Annual International Conference on Mobile Computing and Networking. Seattle, WA: ACM Press (1999) 271-278
29. Tarasewich P., Campbell C., Xia T., and Dideles M.. Evaluation of Visual Notification Cues for Ubiquitous Computing. In Proceedings of 5th International Conference on Ubiquitous Computing, Seattle, WA. 2003