Overseer: A Mobile Context-Aware Collaboration and Task Management System for Disaster Response

Faisal Luqman Carnegie Mellon University Silicon Valley faisal.luqman@sv.cmu.edu

Abstract

Efficient collaboration and task management is challenging in distributed, dynamically-formed organizations such as ad hoc disaster response teams. Ineffective collaboration may result in poor performance and possible loss of life. In this paper, we present an open multi-agent system, called Overseer, that leverages context information in a mobile setting to facilitate collaboration and task allocation for disaster response. We describe our system architecture. deployment, evaluation metrics. challenges and proposed solutions. We also show how mobile context can be used to create dynamic rolebased assignments to support collaboration and effective task management.

1. Introduction

Task monitoring and reassignment in large-scale disaster response, in rapidly changing environments, in the presence of data explosion is challenging, since many events may get lost in such volume of data. Intelligent decisions need to be made quickly, evaluated and changed depending on updated information. In disaster response where the user's environment changes rapidly, mobility context is an essential element that needs to be part of the decision making process. However, there is a lack of systems that continuously monitor the mobility context of the task, and if additional resources need to be added, removed, or reassigned to someone else who could do it faster and within the given time constraints.

There is a clear need to embrace new technology to enhance existing disaster response systems that are illsuited for mobile environments. Government studies [22] have shown that existing communication systems used by emergency response teams are ineffective and incompatible across teams. In recent years, the Martin Griss Carnegie Mellon University Silicon Valley martin.griss@sv.cmu.edu

interoperability of existing emergency response communications has been highlighted [13]. In addition, in some situations, having volunteers assist in disaster response may be desirable in the event that emergency responders are unable to arrive in time; however, the closed communication systems used by first responders and emergency response personnel are often incompatible with those used by average citizens.

Furthermore, existing tools and technologies are inadequate for spontaneously formed disaster response teams. Well-informed decisions need to be made rapidly and reevaluated periodically in order to gauge their effectiveness. Expert advice and opinion may need to be obtained swiftly prior to carrying out a particular task (e.g. determining structural safety of a building before sending team members in to rescue survivors after an earthquake).

In existing first responder communication systems, roles are well defined. Expert or specialist advice is coordinated through centralized control from a single emergency operations center (EOC) or incident command center (ICC). For example, if a team of firefighters encounter an injured survivor, they would send this information to their command center, which would in turn dispatch paramedics to the scene or relay the information to a relevant agency.

In spontaneous disaster response teams comprised of trained first responders and untrained volunteer citizens, roles are ambiguous and not clearly defined. In a disaster response environment with resource and time constraints, available skills and resources both within teams and across teams are often overlooked [5]. A study conducted by IBM indicated that subject matter experts within a corporate organization are rarely consulted [18]. Identifying subject matter experts or specialists within an ad hoc response team in a critical resource constrained environment would be even more challenging.

Ineffective collaboration and task management in disaster response environments may result in possible

loss of life as well as poor performance, which adds to high operation costs and longer disaster recovery time.



Figure 1. Ad hoc disaster response team comprised of first responders and volunteer citizens

Figure 2 by Dynes [10] describes the eight sociotemporal stages of disaster. During Stages 3-5, disaster impact occurs and individuals start forming an ad hoc community to assess the situation and rescue injured survivors. Our system focuses on Stages 3-5 where most loss of life occurs due to poor coordination and task management under time and resource constraints. A system that is capable of overseeing task management could help to ameliorate the losses during these stages.



Figure 2. Eight Socio-Temporal Stages of Disaster, Dynes [10]

In this paper we present an open multi-agent system, called Overseer and explore how context, specifically data and contextual information obtained through sensors on a mobile device, can be leveraged to enable efficient collaboration and task management for disaster response. Our approach uses a multi-agent system architecture supporting heterogeneous devices that enable increased collaboration efficiency, augmented by mobile sensor data (e.g. accelerometer, GPS, audio) and mobility context information.

We also present a scenario that motivates the design of our system: An earthquake leveled a large metropolitan city in California. Many survivors are trapped under rubble and require medical attention. The emergency response team's resources are stretched, given the magnitude of the natural disaster. In order to locate and rescue survivors quickly, the first responders formed an ad hoc response team with untrained citizens who escaped unharmed. The group is divided into two teams, a search team to locate survivors and a foraging team to gather medical and other equipment. Specific tasks are allocated to each team member. To achieve the main objective, both teams must succeed under serious time and resource constraints

The research contributions of this paper are as follows. First, we describe the use of data and context awareness obtained from sensors on mobile devices to facilitate collaboration and task management in a dynamically formed disaster response team. Second, we demonstrate how our system can be used to create dynamic role-based assignments using mobile context awareness and user expertise.

The rest of this paper is organized as follows. Section II describes our system architecture and describes our system design. Section III presents metrics for evaluating our system. We discuss related work in Section IV. In Section V, we discuss the challenges of adoption by existing emergency response teams, usability, security and privacy, along with proposed solutions. Section VI discusses future work and our conclusions.

2. Implementation

In this section, we describe our overall system architecture for mobile, context-aware collaboration and task management. We structured our system as a multi-agent system to allow us to take advantage of the inherent properties in multi-agent systems such as autonomy, sociability, reactivity and pro-activeness. In addition, a multi-agent system model enables dynamic system extensibility and reconfiguration. Additional agents can be easily added, removed or changed. Furthermore, the agent paradigm closely models the inherently chaotic, dynamic and unpredictable nature of disaster scenarios.

While we assume a dedicated centralized disaster response engine in our current system, we envision an ad hoc collaborative system in the future where the disaster response engine is distributed amongst the various agents, running on a heterogeneous set of devices and servers.

2.1. Architecture

Figure 3 is a high-level view of our disaster response system, consisting of an ad hoc disaster response team communicating with a disaster response engine. The ad hoc team combines both untrained citizens and experienced first responders. All members on the team communicate with the disaster response engine through software agents on their mobile devices. In addition to communication, the software agents interact with various sensors on the mobile devices, which are described in more detail in the following section.



Figure 3. Overseer System Architecture

The Overseer system leverages the various sensors and mobile device capabilities to send mobile contextual information about the user to the disaster response engine periodically. The disaster response engine receives and monitors every team member's mobile context, the current task they are engaging in, and calculates the probability of task completion continuously. Depending on the team member's mobile context and calculated probability of task completion, the disaster response engine assigns additional resources to existing tasks and reassigns tasks as needed. If a task is considered to be high priority or if it might cause potential bottlenecks, the disaster response engine will identify and assign additional resources.

The disaster response system allows the task monitoring process to be automated. In large scale disaster response, in which the system size and complexity increases, the number of sensor events generated from the mobile device sensors can potentially overwhelm manually managed command centers or team leaders who oversee the disaster response process.

2.2. Mobile Device Sensors

Our system leverages the following sensors to gather mobile context. In this section, we briefly list some of the sensors we are considering; detailed analysis of their capability, use and calibration is to be discussed in a companion paper.

Accelerometer: The accelerometer is used to determine team members' current activity and task intensity. Using accelerometers to determine a user's activity is an active area of research and is beyond the scope of this paper. We leverage existing work in this area, [15], [1], towards determining team members' mobile context.

Ambient light sensor: An ambient light sensor is used to measure the light intensity experienced by the team member.

Capacitance sensor, Camera: A capacitance sensor is used in conjunction with a front camera with facial recognition to determine a user's level of engagement and attentiveness to the mobile device.

GPS, Compass: A GPS and compass are used to determine a team member's location, direction and general movement speed. We note that GPS may be inadequate near some structures, such as within buildings and underground. Work is being done in a companion research project [14] within our research group to improve interior positioning and firefighter tracking.

Bluetooth: Bluetooth is used to determine the density of team members surrounding the current team member, based on the Bluetooth signal strength of team members nearby. It can also be used for ad hoc communication when network problems arise and for communicating with future Bluetooth capable biometric sensors to monitor responder "health".

Microphone: The microphone on the mobile device is used to capture surrounding sound.

In addition to the above mentioned sensors, the network latency of each team member is monitored. The disaster response engine periodically sends a ping message to each agent on the mobile device. The agents send an acknowledgement packet upon receiving the ping message. The network latency is calculated using the measured round trip time. Observing network latency is important in order to assess the level of communication quality with each team member. Critical tasks should not be assigned to a team member with high latency, as there is a possibility of communication loss, and consequently, loss of the task result. Similarly, tasks already assigned to team members who experience an increase in network latency should be reassigned to another team member, or supported with a team member nearest to the team member experiencing high network latency.

We assume each member of the ad hoc disaster response team has mobile devices with (most of) these capabilities. As smartphones become more powerful, sensor-rich and ubiquitous, we feel that this is a reasonable assumption. See Figure 4; each of these smartphones, as well as similar phones by Samsung, HTC, RIMM and Sony-Ericsson, is well endowed with such sensors. In addition, they have improved battery life and enhanced communication capabilities such as Bluetooth and IEEE 802.11.



Figure 4. Smartphones - Apple iPhone, Nokia N900 and Motorola Droid equipped with rich array of sensors

2.3. Mobile Context

The event data measured by the mobile device sensors described in the previous section provide our system with mobile contextual information. We describe the various components of mobile context collection and their application in our disaster response system below.

Movement Speed: Our system keeps track of every team member's current movement and speed using the GPS. In our initial design, we define current movement speed as average movement speed in past 5 minutes. Note that this number is not fixed, and should be dynamically adjusted depending on context. In addition to the current movement speed, the total average movement speed is also recorded. By doing so, highly mobile team members can be assigned tasks that require scouting or searching with time constraints. The accelerometers can be used to give a much more immediate indication of current activity and movement.

Location and Direction: Location-specific tasks can be assigned to team members who are nearest to the destination. Team members who are headed in a certain direction can be assigned a task along the member's projected trajectory.

Activity and Task Intensity: A user's current activity and level of intensity affects the ability to

perform a given task [3]. High priority tasks are assigned to team members who haven't engaged in a long period of intensive activity. Tasks in progress can be reassigned if a team member's activity intensity reaches a certain threshold, or additional resources could be added to help with the task.

Attentiveness: Tasks that need to be completed with the highest priority are assigned to users with the highest level of attentiveness. We assume that those who are currently engaged with and looking at their mobile devices will immediately receive and respond to the newly assigned task.

Lighting: Tasks that require visual clarity may be assigned to team members with better lighting. Lighting could also be used to determine locations where injured survivors requiring medical attention could be moved.

Network Latency: Figure Network latency is continuously measured. High priority tasks are supported with or reassigned to other team members if the network latency reaches a certain threshold. The team member can also be requested to take measures to improve network latency such as switching locations or moving outdoors if possible. Alternatively, nearby inactive team members can be requested to go to the same location to provide support or to serve as a proxy to preserve communication with the disaster response engine.

Team Density: Critical tasks could be assigned or reassigned to team members with a high concentration of other team members close to them. This allows nearby team members to provide assistance if necessary.

Audio: Tasks that could potentially be affected by noise levels and require a high level of concentration could be assigned to team members with low noise levels.

2.4. Deployment and Initialization

For our current disaster response system design, we assume that the telecommunications infrastructure is intact. However, we acknowledge that this may not necessarily be the case. In real disaster scenarios, communications infrastructures are often destroyed or rendered inaccessible [5]. Thus, in addition to the dedicated server deployment in our proposed system, we also describe possible ad hoc peer-to-peer deployment and communication.

Dedicated Server Deployment: After the formation of an ad hoc disaster response team, each team member will connect to a dedicated server, which also serves as the disaster response engine, using their mobile phone browser. The disaster response engine parses the incoming HTTP request from the mobile

devices and examines the user agent header to determine the device type based on the operating system and platform. For example, an Apple iPhone device has the following user agent information in the HTTP request:

User Agent: Mozilla/5.0 (iPhone; CPU Mac OS X; en)

The disaster response engine then sends the corresponding client agent program binary for each device. After installation, the client agent program launches and displays the user name obtained from the owner's information in the mobile device. The team member can accept or modify the user name and presses next. The agent program then requests the team member for one or more relevant areas of expertise. In Figure 5, we show the list used by our initial system, using the following expertise areas: Structural, Firefighter, Medical and Transportation. This terminology is based on the Department of National Securities Incident Management System [8].

Main Ask Expert			
Please select your expertise level for the following areas:			
Structural	Novice	Intermediate	Expert
Firefighter	Novice	Intermediate	Expert
Medical/Health	Novice	Intermediate	Expert
Transportation	Novice	Intermediate	Expert
Previous			

Figure 5. Disaster response client GUI

Ad Hoc and Peer-to-Peer Deployment: An alternative deployment is to use ad hoc or peer-to-peer deployment. In this scenario, one first responder's mobile device will act as the disaster response engine. Multiple core responders' devices could act as a redundant/collaborating set of engines. In this configuration, other team members in the response team will connect to the first responder's mobile device through Bluetooth to obtain the corresponding client side agent.

2.5. Dependency Graph

The dependency graph, Figure 6, is a tree structure that captures the dependency between various subtasks that are needed to accomplish a larger objective. The dependency graph is used to calculate the probability of a task succeeding.

To create the dependency graph, the main objective is decomposed into smaller sub-tasks, which are then further sub-divided into individual tasks amongst team members. The number levels in the dependency graph correspond to the number of times the objective is subdivided, depending on the size and complexity of the main objective. Automated task decomposition [19] is beyond the scope of this paper. In our initial design, we assume that the process of dividing tasks is done manually by the disaster response team leader.

In addition to displaying names of team members who are assigned tasks, the dependency graph also shows team members who have not yet been assigned tasks, as well as expert feedback on the task at hand. For example, a team member tasked with gathering medical equipment will have a better idea of what items to gather with guidance from a medical expert compared to a team member who did not receive any expert advice.

2.6. Probability of Task Completion

Each node in the dependency graph is assigned a success probability value. The probability value is determined by the mobile context as well as other factors such as external monitors and expert input.

If the probability value for a given node drops below a threshold, the disaster response engine alerts the team lead about a possible critical issue and takes steps to increase this value above or as close to the threshold as possible. This may be done by assigning additional resources, such as idle team members nearby, that will give the highest impact to the tasks. For example, to assist with the task of finding medical supplies, team members who are closest to the search site (location) and fast (movement speed) may have a higher impact on the probability value. The success probability value based on mobile context is described below.



Figure 6. Task dependency graph and task completion probability

Individual Tasks: The probability of success for a task assigned to a single individual is determined by

the mobile context relevant to the specific tasks such as movement speed, location, activity intensity, attentiveness, lighting, network latency and audio, illustrated in Figure 7.



Figure 7. Mobile context used to determine probability for individual tasks

The effect of specific mobile context will vary according to the task being performed. For example, if a team member is foraging, searching or carrying items, movement speed and location will carry more weight. If the team member is tasked with observing and reporting critical events, network latency, lighting and audio would be more important. In the default case where the effect of specific mobile contextual information for certain tasks cannot be determined, all mobile contextual information will be weighted equally.

Overseer currently will use a rule-based system to determine the weights of different mobile contextual information for different sub-tasks; more work to create a powerful statistical learning system is planned.

Collective Tasks: For collective tasks involving two or more team members and optional expert input, weights are assigned to each individual sub-task as well as to expert input. The weights, shown in Figure 8, reflect the importance of a sub-task to ensure completion of the larger tasks. In the case of expert input, the weight reflects the level of expert knowledge. For example, expert input carries more weight than an intermediate or novice input. The weights could be adjusted dynamically as team members perform and complete assigned tasks, similar to reputation systems [12]. For simplicity, we assume that all sub-task weights and expert input are equal in our initial design.



Figure 8. Weights assigned to various sub-tasks and expert input

2.7. Dynamic Task Allocation

An important aspect of disaster response is to dynamically adapt assigned tasks to the changing environment. There has been numerous papers on task and role allocation. Existing methods include negotiation where agents place bids on available tasks, with a mediator assigning the task to the highest bidding agent [23].

In Overseer, the response engine continuously monitors and assigns tasks based on each team member's mobile context values using a rule engine. For example, if an assigned scout's movement speed falls below a threshold and stepped out a given radius of a search location, Overseer attempts to add or reassigns the task to team members who fit the criteria in the rule engine.

2.8. Real Use Case Scenarios

We present real use case scenarios to show the practicality of Overseer.

Scenario 1: Team members 1 and 2 from sub team B are tasked with gathering medical equipment to treat survivors. Overseer detects that Team member 1 has previously engaged in prolonged high intensity activity and is moving slowly. The probability for team member 1 task completion decreases, thereby affecting the overall probability of successfully completing the foraging task. Overseer adds an idle team member from sub team A, who has a high movement speed and is close to the foraging location to the task of foraging to increase the probability of success for gathering medical equipment.

Scenario 2: Team member 5 from sub team A is treating a survivor in a partially collapsed building. The network latency for team member 5 gradually increases. Overseer directs nearby team members towards team member 5 to serve as a proxy for communication. A fire begins to appear in an adjacent structure and starts to spread towards the building where sub team A is located. Scouts located nearby alert Overseer of the fire, which in turn sends an alert to team members with the highest attentiveness value.

2.9. Expert Input

We designed our disaster response system to take advantage of existing expert knowledge within the ad hoc disaster response team itself. When tasks are being carried out, individual team members can query the disaster response system with specific questions in the Ask Expert tab in Figure 5. The disaster response system will attempt to match the query with an expert who is in the best position to provide feedback. For example, before entering a partially destroyed building to search for survivors, team members can send a query asking if it is safe to enter the building. The system will locate experts with structural expertise who are close by, or relay an image of the building structure to experts.

2.10. System Overview

The information flow in the mobile, context-aware disaster response system is shown in Figure 9. First, the main objective is decomposed into several smaller sub-tasks. The sub-tasks are further decomposed into smaller tasks until they are assigned to individual team members to form a task dependency graph.

The task dependency graph is sent as an input to the disaster response engine, together with expert feedback and continuous mobile context obtained from each team member's mobile device.



Figure 9. System overview: Information flow

The process then becomes a continual loop of monitoring tasks based on each team member's mobile context and taking specific actions based on the mobile contextual information, such as reassigning task and assigning additional resources.

2.11. Multi-Agent System

We implement our work as a multi-agent system. A software agent is an autonomous and loosely coupled software component responsible for a logical grouping of functionality, operating in a heterogeneous computing environment, and interacting with other agents to produce a multi-agent system.

The use of software agents enhances flexibility and enables dynamic evolution. A software agent architecture was chosen for a variety of reasons, including: 1. Agents represent a convenient architectural metaphor, system design and implementation choice because of their agile, dynamic, loosely coupled component structure. Agents support principles of modularity, high cohesion and loose coupling, and independent component development and evolution.

2. Agents support runtime discovery and configuration, similar to, but richer than, object-oriented modules and web services.

3. Over the last few years, several robust and widely used open source agent toolkits such as JADE/LEAP [4] and SemanticAgent have been developed, which run on servers, laptops and personal devices. Standardized protocols allow inter-operation between agents and services, and between several different agent toolkits. Several extensions exist that support workflow, intelligence and machine learning.

4. Small devices have grown in memory and computational capacity, enabling them to be used as full-fledged peers in a heterogeneous agent environment.

5. Agents are good for mixed-initiative agile and (semi-) autonomous systems. As agent capabilities grow, humans can delegate responsibilities to agents. Furthermore, agents and humans can engage in relatively natural dialog and negotiation.

3. Evaluation

In order to evaluate the effectiveness of our system, we propose two different metrics, situational and effectiveness. The situational metric measures the phase and state of the collaboration, while the effectiveness metric measures the success of the collaboration and task management itself.

Situational Metric: The state and phase of the collaboration is captured by the situational metric. We refer to the forming-storming-norming-performing model of group development [21] in describing the situational metric. Our system solicits team members for measures of their expertise on several key areas and uses this knowledge in addition to their mobile context to match queries with experts who could provide the best input and advice. Thus, our system design accelerates the transitional phases of collaboration towards the performing stage by facilitating the storming and norming process. In the future, we plan to use expert knowledge information to assist the forming phase, as well as statistical analysis using mobility context history to augment the performing phase.

Effectiveness Metric: The effectiveness metric measures the success and progress of the collaboration and task management. We compare the performance

of our system against a baseline collaboration environment that does not take mobile, context awareness into account. Specifically, we compare the speed in completing the main objective, the number of failed sub-tasks, and the overall effort expended by the disaster response team. We believe that our disaster response system will perform better than the benchmark by taking advantage of expert knowledge within the disaster response team and by ensuring that the probability of success of sub-tasks is above a certain threshold through continuous task monitoring and mobile contextual information analysis.

To validate our system, we first plan to conduct simulations using the JADE framework to fine tune various mobile context parameters. We are building a first prototype using the JADE framework implemented on Nokia N800 mobile devices running the Maemo 2008 operating system. We then plan to test this prototype using a simulation of firefighters and responders in the DART collapsed structure testbed at NASA Ames Research Center [7].

4. Related Work

Palen et al. [16] describes the emergence of information pathways for public participation in disaster response, and discusses the need to coordinate improvised activities between temporary organizations and formal response teams. It includes examples of how the majority of victims in disaster are saved by local, ad hoc volunteer groups. Our scenario is informed by their findings.

Barnard et al. [3] discusses how various contextual conditions such as motion, lighting and task type affect a user's performance and workload. We leverage their findings, and extend the use of mobile context further by incorporating movement speed, location, direction, activity intensity, attentiveness, lighting, network latency, team density and audio into our system.

Beep [11] is a collaboration-oriented design of a disaster response system. Beep uses mobile devices equipped with GPS for the coordination of evacuation procedures for large groups of victims. Similar to Overseer, Beep uses a central server to monitor user context, such as location, direction, stress and experience in device operation, and informs standby rescuers when assistance is required. Overseer expands on this by continuously monitoring tasks being performed, evaluating various mobile contexts and dynamically assigning and reassigning tasks to disaster response team members.

Bahl et al. [2] uses an inference graph to capture various dependencies in an enterprise network and a fault localization algorithm to determine the root cause of the failure. We apply a similar approach to the disaster response domain, with several modifications to take into account mobile context. The use of various sensors on mobile devices as well as possible external monitors is also added to the decision process. Our system is proactive in determining bottlenecks, and identifies measures to address problems.

Singley et al. [18] describes an expertise-sharing system that connects question askers with subjectmatter experts using synchronous chat in a globally distributed corporate environment. Overseer applies the expertise sharing concept in disaster response. However, Overseer uses asynchronous communication and dynamically selects subject matter experts by assessing their mobile context.

Inferring physical mobility states from sensor data (such as accelerometers) has been an active area of research. We leverage existing research in this area, [15], [1], and apply them to our system to determine the type and level of user activity.

Many other researchers are exploring the use of single and multiple sensors on mobile devices; this includes companion work in our research group at Carnegie Mellon Silicon Valley: [6], [20], [1] and [14].

5. Discussion

In general, we are satisfied with our initial design. To get to this point in the design, we placed several constraints on our initial system, which must be addressed to produce a system capable of working in a more realistic environment. For the large part of our system design, we assume that the various client agents on mobile devices communicate with the disaster response engine through the existing communication infrastructure. In large scale disasters, the communications infrastructure often collapses and is rendered ineffective, as was the case during Hurricane Katrina. While we briefly described an alternative deployment and communications strategy using ad hoc and peer-to-peer communications in hastily formed networks, we plan to investigate this scenario more thoroughly in the future to address this limitation.

We also assume that the ad hoc disaster response team has already been formed a priori. In the future, we plan to add features in the disaster response system, such as anonymous voting, and to use the expert characterization coupled with mobile context to augment the initial formation of the disaster response team. This allows the formation of virtual, collaborative, ad hoc disaster response teams.

Mobile devices are also constrained by limited battery life. Radio communication consumes a significant amount of power. Thus, unnecessary communication should be reduced in order to conserve battery life. In the future, we plan to use the agent program on the mobiles to determine and send only significant mobile context information to the Overseer disaster response engine using statistical analysis.

There are several potential issues that arise concerning the integration between the existing system and Overseer. Among these are adoption, usability, security and privacy issues. The issues of adoption and usability are significant. While many trained first responders rely on well established systems and procedures, other people are more willing to try out new technologies such as Twitter in the event of disasters or emergencies [17]; also, prior crisis experience motivates people to use new technology. Thus, we can design extra capabilities into the system with a positive expectation on (eventual) adoption and usage.

The usage of mobile devices in dynamically changing and critical scenarios can be challenging. For example, smoke from fires obscure vision, and small interfaces can be difficult to read while running or performing intensive tasks. The interface should be designed to make it easy to interact with when wearing gloves and incorporate visual and tactile feedback in order to be effective in noisy environments.

The issue of privacy vs. emergency is an interesting topic. In a disaster response environment, we believe victims may be willing to give up certain privacy information, such as location. Similarly, existing members of the ad hoc disaster response team may also be willing to give up certain aspects of privacy to preserve their safety while they attempt to rescue survivors and address the disaster at hand.

We argue that for natural disasters, the incentive for malicious or rogue users to compromise the security of the disaster response system is low. However, this may not be the case for disasters triggered by terrorist attacks. Thus we acknowledge that our existing system design is domain specific and limited in scope. In the future, we plan to enhance the security aspects of the mobile context aware disaster response system to make it more suitable for a broader range of applications.

6. Future Work and Conclusions

In this paper, we describe Overseer, a mobile context-aware collaboration and task management system for disaster response that monitors and dynamically manages tasks assignment based on team members' mobile context as well as feedback from experts within the response team.

In designing Overseer, we make two important contributions. First, we describe the use of data and

context awareness obtained from sensors on mobile devices to facilitate collaboration and task management in a dynamically formed disaster response team. Second, we demonstrate how our system can be used to create dynamic-role based assignments using mobile context awareness and user expertise.

We plan to make several improvements in Overseer. In our current design, the task dependency graph is generated manually. In the future, we will investigate methods to automate this process using keyword analysis between team members. As complexity and number of dependencies increases, automated task dependency graph generation will be an important issue to ensure that our system scales well.

We also intend to improve the assignment of weights for the task success probability calculation. In the current design, we use a simple rule-based system to determine the weights based on different sub-tasks. We plan to augment this using statistical analysis by analyzing the history of similar tasks and mobile contexts.

Finally, we plan on incorporating statistical analysis of user behavior to determine what time of day they are more active, and to assign tasks based on activity level.

7. Acknowledgements

We'd like to thank Patricia Collins, Wendy Fong, Estelle Dodson, Michael Sims, Alec Dara-Abrams and Senaka Buthpitiya for reviewing our paper and providing valuable advice. We also thank the anonymous reviewers for their comments and feedback.

8. References

[1] O. Abdul-Baki, J. Zhang, P. Zhang, M. Griss, T. Lin. A Mobile Application to Detect Abnormal Patterns of Activity, R. Montanari, P. Zerfos, General Chairs, MobiCASE 2009.

[2] P. Bahl, R. Chandra, A. Greenberg, S. Kandula, D. A. Maltz, M. Zhang. Towards Highly Reliable Enterprise Network Services via Inference of Multi-level Dependencies, J. Murai, General Chair, SIGCOMM 2007, p. 13-24.

[3] L. Barnard, J.S. Yi, J. A. Jacko, A. Sears. Capturing the effects of context on human performance in mobile computing systems, P. Thomas, editor, Personal and Ubiquitous Computing, 2007, p. 81-96.

[4] F. Bellifemine, G. Caire, A. Poggi. JADE – A Java Development Framework, R. H. Bordini, M. Dastani, J. Dix, A. Seghrouchni, editors, Multi-Agent Programming: Languages, Platforms and Applications, 2005, p. 125-148. [5] L.K. Comfort, T.W. Haase. Communication, Coherence, and Collective Action: The Impact of Hurricane Katrina on Communications Infrastructure, R. D. Bingham, C. L. Felbinger, editors, Public Works Management & Policy, 2006, p. 328-343.

[6] H. Z. Cheng, M. Griss. OmniSense: Collaborative Mobile Sensing for Context-Aware Applications, A. Dalton, General Chair, submitted to HotMobile 2010.

[7] Disaster Assistance and Rescue Team (DART), http://dart.arc.nasa.gov/.

[8] Department of Homeland Security, National Incident Management System, 2008.

[9] A.K. Dey, G.D. Abowd. Towards A Better Understanding of Context and Context-Awareness, T. Turner, G. Szwillus, M. Czerwinski, F. Peterno, S. Pemberton, editors, CHI 2000.

[10] R.R. Dynes. Organized Behavior in Disaster, 1970.

[11] L.T. Gunawan. Collaboration-Oriented Design of a Disaster Response System, M. Czerwinski, A. Lund, General Chairs, CHI 2008, p. 2613-2616.

[12] D. Kostoulas, R. Aldunate, F. Pena-Mora, and S. Lakhera. A nature-inspired decentralized trust model to reduce information unreliability in complex disaster relief operations, J. C. Kunz, T. Tomiyama, I. F. C. Smith, editors, Advanced Engineering Informatics, 2008, p. 45-58.

 M. Kyng, E.T. Nielsen, M. Kristensen. Challenges in designing interactive systems for emergency response, J. M. Carroll, S. Bødker, J. Coughlin, editors, DIS 2006, p. 301-310.

[14] H. Lin, Y. Zhang, M. Griss, I. Landa. WASP: An Enhanced Indoor Locationing Algorithm for a Congested Wi-Fi Environment, R. Fuller, X. Koutsoukos, Y. Zhang, editors, MELT 2009.

[15] E. Miluzzo, N. Lane, K. Fodor, R. Peterson, S. Eisenman, H. Lu, M. Musolesi, X. Zheng, and A. Campbell. Sensing meets mobile social networks: The design, implementation and evaluation of the CenceMe application, T. Abdelzaher, General Chair, SenSys 2008, p. 337-350.

[16] L. Palen, S. Liu. Citizen Communications in Crisis: Anticipating a Future of ICT Supported Participation, M. B. Rosson, General Chair, CHI 2007, p. 727-736.

[17] I. Shklovski, M. Burke, S. Kiesler, R. Kraut. Technology Adoption and Use in the Aftermath of Hurricane Katrina in New Orleans, Position paper for the HCI for Emergencies Workshop, CHI 2008.

[18] K. Singley, J. Lai, L. Kuang, J.M Tang. Bluereach: Harnessing Synchronous Chat to Support Expertise Sharing in a Large Organization, M. Czerwinski, A. Lund, General Chairs, CHI 2008, p. 2001-2008.

[19] P. Stone, M. Veloso. Task decomposition, dynamic role assignment, and low bandwidth communication for real-time strategic teamwork, H. Kitano, editor, Artificial Intelligence, 1999, p. 241-273.

[20] L. Sun, P. Collins, T. Selker, M. Griss. PainSense: Pain Assessment Through Reality Sensing, A. Dalton, General Chair, submitted to HotMobile 2010.

[21] B. Tuckman. Forming-storming-norming-performing, www.wikipedia.com.

[22] U.S. General Accounting Office. GAO-04-494 Project SAFECOM, Key Cross-Agency Emergency Communication Effort Requires Stronger Collaboration, April 2004.

[23] A. Campell, A. S. Wu, Task and Role Allocation Within Multi-Agent and Robotics Research, Technical Report CS-TR-07-05, University of Central Florida, EECS, 2007.