LocalAlert: Simulating Decentralized Ad-hoc Collaboration in Emergency Situations

Silvia Nittel, Christopher Dorr, and John C. Whittier

Spatial Informatics, School of Computing and Information Science
University of Maine, Orono, USA
`nittel@spatial.maine.edu`, `christopher.h.dorr@maine.edu`,
`john.c.whittier@maine.edu`

Abstract. Today, advances in short-range ad-hoc communication and mobile phone technologies allow people to engage in ad-hoc collaborations based solely on their spatial proximity. These technologies can also be useful to enable a form of timely, self-organizing emergency response. Information about emergency events such as a fire, an accident or a toxic spill is most relevant to the people located nearby, and these people are likely also the first ones to encounter such emergencies. In this paper we explore the concept of decentralized ad-hoc collaboration across a range of emergency scenarios, its feasibility, and potentially effective communication protocols. We introduce the *LocalAlert* framework, an open source agent simulation framework that we have developed to build and test various form s of decentralized ad-hoc collaboration in different emergency situations. Initial experiments identify a number of parameters that affect the likelihood of a successful response under such scenarios.

Keywords: decentralized spatial computing, decentralized ad-hoc collaboration, emergency situation management, agent framework, ad-hoc communication, ad-hoc communication protocols

1 Motivation

Technological advances in mobile phones, location-based social network applications, and ad-hoc communication abilities will change the ways in which people respond to emergency situations in the future. Emergencies vary greatly, from far reaching events such as fast moving wild fires, hurricanes, or flooding, to events experienced on a smaller spatial scale such as a bomb threat in public building, an assailant at a school or university, or an accident at a local chemical plant. These events are characterized by occurring suddenly, requiring immediate reaction, and being first encountered by people in close proximity to the event.

In the domain of geosensor networks, the term decentralized spatial computing was coined [13] to capture the fact that while individual sensor nodes only capture a local glimpse of a geographically larger phenomenon, they can collaborate with their immediate local neighbors to identify a larger phenomenon. In this context, global control or coordination is not necessary, nor do local nodes need to understand the global phenomenon to coordinate locally. The paradigm of ad-hoc situational collaboration could be similarly powerful in emergency response situations that involve information related to spatio-temporal events and the people located their proximity.

Imagine the following scenario: Mary is shopping at a local superstore. While she sorts through some items on a shelf, she hears a person screaming and a shot being fired. She can determine the general direction of the sound, but she cannot find out what is really happening. Alarmed and scared, Mary checks her smartphone application, *LocalAlert*, a real-time location-based spatial event notification and coordination application. *LocalAlert* is location-aware, and enables short-range communication between people in spatial proximity without the need for users to know each other or connect to a centralized infrastructure. *LocalAlert* recognizes Marys location and can detect other people in her proximity. The application might then display (via text or graphics) information about the already-known event in the store. If not, Mary can ask a question that is forwarded to others in her proximity and ultimately relayed to people who may have encountered the event first hand. Mary is now better informed and decides to leave the store using a safe route. Mary continuously checks *LocalAlert* for updates in case the shooter has moved, and evaluates various escape routes. Using *LocalAlert*, Mary can retrieve up-to-date information about the situation as provided by other people in the same emergency. Other scenarios include a bomb threat in a public building or apartment complex, or a fast moving wildfire. *LocalAlert* can help to identify shortest evacuation routes for people unfamiliar with a building floor plan, or display notifications about blocked routes by other people who encountered them. *LocalAlert* is not restricted to emergency situations either; it can also be useful in other location-based proximity scenarios. For example, drivers might be stuck in a suddenly occurring traffic jam and want to know the length of the traffic jam or its cause.

Today, several technical and non-technical challenges remain. While GPS can be used for outdoor localization, determining accurate indoor location is still an active research area; this is relevant if *LocalAlert* is combined with mobile mapping. Further, todays mobile phones and smartphones have limited ad-hoc communication abilities based on short-range radio devices and ZigBee-based mesh networks [21]. This, how-ever this is changing rapidly due to the advantages of secure short-range communication enabled by ZigBee. User interface questions also remain: in which form should information about events be created? It might take too long to type in textual event messages in time-critical emergency situations, and text-based messages are difficult to automatically aggregate.

Decentralized self-organizing applications for emergency situations do not (yet) exist. Our first objective is to test the general feasibility of such an approach. Our second objective is to investigate different ad-hoc communication protocols and coordination strategies between smart device users to identify effective protocols under different circumstances. For example, if users already have partial event information, they might prefer to pull (*ask*) for additional information. On the other hand, people first discovering a suspicious event will likely alert (*tell*) other, unaware people in their vicinity who could be affected. Or perhaps some combinations of the two approaches would occur? Which communication protocol leads to information saturation in the system quicker? What are the key parameters of such an information dissemination system, and how do these parameters depend on each other? Beside decentralized *notification* of an event, can *collaborative coordination* also be achieved? To investigate the feasibility and limitations of ad-hoc decentralized coordination we have implemented the *LocalAlert* simulation framework. In *LocalAlert*, smartphone users are modeled as agents in a spatial environment in which they follow routes and accomplish objectives. The simulation environment accommodates both indoor and outdoor spaces, a rich set of dynamic event types, and a range of ad-hoc communication protocols and coordination strategies. We tested them under varying input conditions, such as different agent populations, event types, and behaviors, and tested the efficiency of the notification and coordination strategies.

The remainder of this article is structured as follows. In Section 2, we describe the technological background of this research to demonstrate feasibility. Section 3 introduces our *LocalAlert* simulation environment and Section 4 contains our performance analysis. Section 5 discusses related work, and Section 6 offers our conclusions and identifies possible future work.

2 Background

In this section, we present the background consisting of different research areas and technologies that are related to our exploratory approach. We also briefly discuss the state of the art in these areas to assess the feasibility of our approach.

2.1 Ad-hoc Communication Technology

Ad-hoc communication technology [7] has been available for several decades and has found widespread application in *wireless sensor networks, mesh networks* and *mobile ad-hoc communication networks* (MANETs). Instead of relying on preexisting routing infrastructure with routers or access points, a wireless ad-hoc communication network is decentralized. Here, all network devices have equal status and can connect with any other devices in their wireless link range. The communication topology of the network is built *ad-hoc* based on node proximity, availability, and wireless link properties, and devices participate in the *routing* of messages by forwarding data to other more distant nodes via multiple “hops” (see Figure 1). The routing methods in ad-hoc networks attempt to *dynamically* find paths between two nodes A and B. The presence of dynamic and adaptive routing protocols make it possible to set-up ad-hoc networks quickly, with minimal configuration, and enable dynamic restructuring on-the-fly since the devices do not need to be known by name ahead of time. This fact makes them well suited for use in a wide range of emergency situations, including natural disasters or military conflicts.

Today, most mobile phones support several types of wireless communication, such as communication over cellular, Wi-Fi and Bluetooth networks. Efforts to provide built-in support for ad-hoc networking in mobile phones are also taking place. For example, ZigBee [21], a widely used interoperability standard specification for various ad-hoc networking protocols, includes the *ZigBee Telecom Services* standard [22] for value-added services such as mobile gaming, secure mobile payments, and mobile advertising. Also, the ZSIM card has been proposed, which provides local ad-hoc communication using the ZigBee mesh protocol and supports local ad-hoc links over distances up to 70m indoors and 400m outdoors. Ad-hoc communication using mobile phones, however, should not be confused with cell broadcast [6] for GSM-based mobile phones. Cell Broadcast (CB) messaging is a one-to-many, geographically focused messaging service that allows users to broadcast a text message anonymously and simultaneously to all phone subscribers currently located within a cell of the larger cellular network. This service however is not available to the average subscriber.

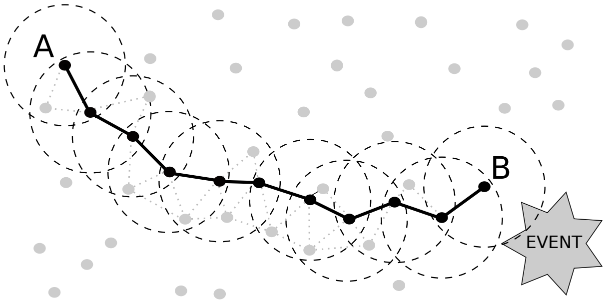


Fig. 1. Ad-hoc communication

2.2 Ad-hoc Collaboration

Ad-hoc communication enables ad-hoc collaboration between computing nodes. It has been introduced as a robust and flexible paradigm by wireless sensor networks to cooperatively accomplish tasks [5]. Ad-hoc collaboration is a higher-level concept than ad-hoc communication; it can be defined as the collaboration of several computing nodes located in spatial proximity to achieve a task, even in the absence of previous communication or collaboration. Since there is no central coordinator with global knowledge that assigns roles or partial tasks to the nodes, every node can act to initiate collaboration and decide to participate in collaboration initiated by other nodes. This mode of collaboration is also called decentralized collaboration. In geosensor networks, ad-hoc collaboration has been used to aggregate locally sensed information collaboratively into global knowledge about a phenomenon such as establishing its currently estimated boundary [8, 9]. In mobile geosensor networks, sensor nodes participate in ad-hoc collaboration with nodes that they encounter in their spatial proximity while moving, and then pass on information to them. Examples of this include vehicles communicating about hazardous road conditions [14] or information exchanges in intelligent transportation networks [16].

2.3 Smartphones and Emergency Management

Today, smartphones account for about a third of the mobile phone market [12]. They are often equipped with GPS, and enable location-based social network applications such as *Foursquare*, *Gowalla*, and *Google Latitude*, which allow users to "check in" to places in real-time. Other location-based social applications include mobile friend finders, mobile gaming applications, and dating applications. Although today, people use social media applications on smartphones to share their location (and potentially other) information in real-time, ad-hoc communication based applications for ad-hoc collaboration do not (yet) exist.

Smartphones and similar mobile devices are also currently used for emergency management. Applications exist that let people store "in-case-of emergency" data on their smartphone – such as critical contact information, a list of current health care providers, or severe allergies – for easy access. Additionally, smartphones and similar devices are used as platforms for centralized updates about emergencies by cities and states. For example, Cupertino, California launched an emergency application that acts like a Rolodex with critical information in case of an emergency (such as earthquakes, wildfires, etc.) with real-time weather and hazard alerts, as well as with meeting place and shelter locations [3, 10]. Similar applications are available in other cities.

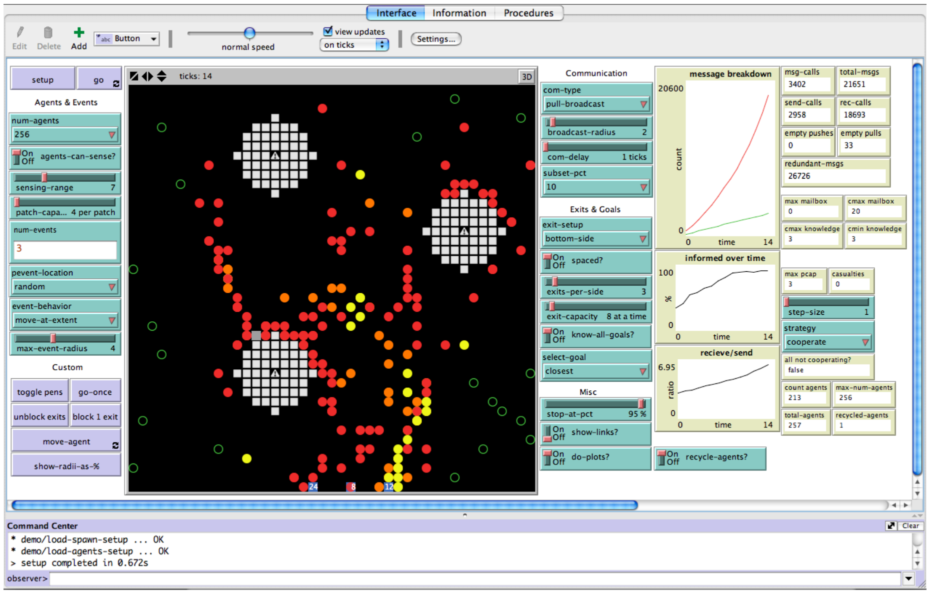
3 Simulating Agent-based Decentralized Ad-hoc Collaboration in Emergency Situations

In this section, we describe the important components of the *LocalAlert* simulation framework, and specify the problem space we have investigated, implemented and tested. To enable modeling of ad-hoc collaboration in emergency situations we conceptualized different types of *space*, *agents* and spatial emergencies (called *events*). Furthermore, we modeled *ad-hoc collaboration strategies* between agents in detail. Our objectives are to firstly investigate the feasibility of this approach and secondly, to identify which collaboration protocols work well under which circumstances.

3.1 Modeling Space and Events in *LocalAlert*

In the *LocalAlert* framework the space serves as a shared environment for all agent entities.

Shared, dynamic space: We provide a base space, represented by an adjustably sized two-dimensional grid of cells, on which entities like agents and events exist. The space is configurable as a combination of freely navigable spaces and obstacles, thus, supporting the modeling of a wide range of indoor or outdoor spaces of varying complexity. The space is composed of patches, which are either *non-agent-barrier* patches, which act as freely navigable space for agents, or *agent-barrier* patches, which represent cells an



num-agents: 256

agents-can-sense?: On

sensing-range: 7

patch-capa... 4 per patch

num-events: 3

pevent-location: random

event-behavior: move-at-extent

max-event-radius: 4

com-type: pull-broadcast

broadcast-radius: 2

com-delay: 1 ticks

subset-pct: 10

exit-setup: bottom-side

spaced?: On

exits-per-side: 3

exit-capacity: 8 at a time

know-all-goals?: On

select-goal: closest

stop-at-pct: 95 %

show-links?: Off

do-plots?: On

recycle-agents?: On

msg-calls: 3402

total-msgs: 21651

send-calls: 2958

rec-calls: 18693

empty pushes: 0

empty pulls: 33

redundant-msgs: 26726

max mailbox: 0

cmax mailbox: 20

cmax knowledge: 3

cmin knowledge: 3

max pcap: 3

casualties: 0

step-size: 1

strategy: cooperate

all not cooperating?: false

count agents: 213

max-num-agents: 256

total-agents: 257

recycled-agents: 1

```
* demo/load-spawn-setup ... OK
* demo/load-agents-setup ... OK
> setup completed in 0.672s
observer>
```

Fig. 2. The *LocalAlert* framework simulating agents and events

agent may not travel over (e.g. event physical barriers and exits). All patches have a patch-capacity. The *LocalAlert* framework supports the use of more complex base-maps; however, they are not used in our current analysis.

Events: Events are defined as an additional but separate agent entity class. Events possess attributes and follow rules which determine their overall behavior: since we are modeling dynamic spatio-temporal events, these behaviors include properties such as maximum expansion radius and rules which dictates how moving events act when they reach a wall. For simplicity, agent-event interaction is limited by the following rules:

1. Agents may freely and safely observe events at any distance greater than one patch. This helps agents in the space who are trying to maintain both their intended heading and some desired buffer distance from all proximate events.
2. Agents and events cannot safely occupy the same location in space at the same time, and agent safety is currently a binary property: completely unharmed (safe), or completely injured (consumed by the event). This reduces model and framework complexity while still producing viable data as it relates to the effectiveness of a given strategy.

3.2 Modeling Agents in *LocalAlert*

Agent entities are modeled around mobile smart-device users, who all have the common goal of obtaining and then disseminating information about spatio-temporal events occurring in the shared space. Agents can sense events directly based on a sensing-range parameter, or via the exchange of information with other agents in the space (limited by a communication-range parameter). They distinguish between two objectives: either wandering freely or responding to an event, depending on the types and amount of information they currently possess. In the default wandering state, agents move freely around the currently unconstrained space and retain this state until they either decide to exit on their own, or encounter or are informed of an event. In either case, when agents switch into the second state of responding, their primary objective becomes quickly but safely exiting the space. The range of an agent's abilities to communicate and sense is determined primarily by the framework parameters. Furthermore, in case of encountering an event first-hand, agents are tasked to perform the collective process of decentralized emergency coordination and self-organization.

3.3 Ad-hoc Communication and Collaboration Strategies in *LocalAlert*

Since communication is central to many of the questions related to this research, *LocalAlert* features an extensive set of communication protocols that determine how agents can communicate with each other. Communication involves agents exchanging messages that contain a variety of spatial information about events and the space it-self. In the simplest form, a message consists of a unique message ID, a message body, and a location, which refers to the location of the spatial entity that the message refers to (i.e. the event). Thus, updates about event locations can be accommodated in this model. Additional message fields, such as the number of hops a message has taken before being received, are also maintained.

Communication: We have currently implemented two communication types: *push* (agents send information) and *pull* (agents request information) strategies. Additionally, for each communication type we investigate two message distribution models, *epidemic* and *broadcast*. Under the epidemic message distribution model, the number of other nodes selected for communication varies from 10% to 90% of the available neighbors in communication range, while in the broadcast mode, a message is flooded across the network (i.e. each agent receiving a message forwards it to *all* other agents in its own communication range).

Collaboration: Collaboration consists of *notification* and *adaptive coordination*. In the notification mode, agents simply request information about an event or inform others about an event. In the collaborative mode, agents exchange spatial information about the *space* (e.g. which exits are blocked?) and the *event* (does the event change location? Where is it now?). Both modes are forms of dynamic collaboration, where old messages about the same event can be updated with new information. We compare three different levels of agent collaboration:

1. *Sensing only:* in this mode, agents do not cooperate or communicate with other agents in the space in any way; all information is obtained solely through an agents sensory capabilities. Thus, the agent has to

encounter the event. This basically reflects todays situation, where adhoc communication enabled smartphones are not used.

2. *Selfish:* in the selfish mode, agents collaborate only until they have fulfilled personal exit requirements, at which point they no longer actively participate in communication with other agents, though they may still act as intermediate nodes in forwarding information to others.
3. *Cooperate:* agents collaborate fully with others for the full duration of their time in the space.

Agent communication is modulated by several framework parameters, i.e. how often agents communicate, with how many other agents an agent communicates, and how many messages an agent may store.

During the communication and collaboration processes, agents perform intelligent message aggregation. Currently, agents combine messages based on the identification of an existing message with matching *location* and *message body* fields. This, along with the other various message fields, allows agents to rank information in a number of ways: for example, an agent may sort all known exit locations by proximity, or by the number of times the agent has received a message using the *times-heard* field. The *LocalAlert* framework also provides additional message handling and decision-making support mechanisms for an agent, such as managing a blacklist, which contains messages that are no longer suitable to pass on, e.g. messages about a previously known safe exit, which is then discovered to be blocked. Upon discovering any such invalid information, agents purge all matching messages from their current message and knowledge lists, with the intent of ensuring out-of-date information is no longer spread.

3.4 Implementation

The *LocalAlert* framework is implemented in NetLogo [19], a free, cross-platform, programmable multi-agent modeling environment. NetLogo is particularly useful for the investigation of models that have dominant spatial or temporal elements, or systems models, which evolve over time. Our strategy was to encapsulate the newly developed core functionality into small, purpose-built modules so that the *LocalAlert* framework is extensible, reusable, and easily expandable to accommodate new functionality without changes to existing code. The code is available at `http://code.google.com/p/gaem/`.

4 Performance Evaluation

4.1 Test Parameters

Our interest is in identifying optimal communication and coordination strategies under a variety of emergency situation scenarios. We constructed a series of nine experiments, representing different combinations of event and response components. These experiments are designed to investigate the influence of the following parameters on our proposed response models: *agent population* or density

(256 vs. 512 vs. 1024 agents), *coordination strategy* (cooperate, selfish, or sensing only), *communication protocol* (push vs. pull-based, broadcast vs. epidemic), *event type* (single fixed, single expanding, single moving, multiple events), and initial *event location* with regard to exit locations. Results from each run are ranked according to the metric "*ticks-to-completion*" which represents the number of iterations required to reach an exit criterion. Ticks to completion serves as the most telling measure of effectiveness since the faster agents are informed, the faster they can make informed decisions and exit. However, as time is not the only measure of effectiveness, we also examine the robustness of a response strategy, as it relates to how likely the strategy is to succeed.[1]

4.2 Validating Decentralized Ad-hoc Collaboration

Table 1 shows the results for an *expanding* event scenario. We evaluated two criteria: *speed* (minimum ticks-to-completion) and *reliability* (how likely a given strategy is to succeed). For each agent population, a total of four columns are presented: the first two columns represent the top 10th percentile of successful runs (raw count and percentage), and 3rd and 4th column represent the number of successful runs per strategy and its percentage of the overall runs. For example, we simulated 720 runs with 256 agents, and 188 of the successful runs used the full cooperation strategy, while 170 runs used either the selfish or sensing only strategies. Additionally, for each agent population, a pass rate (PR) value is provided, representing the total pass rate (number of successful runs/number of runs tested) regardless of configuration: for example, roughly 73% in the case of an expanding event with 256 agents. For expanding events, the tests show that the full cooperation strategy results in the fastest exiting of agents, with this strategy representing 60-76% of the fastest successful runs. The relative

Table 1. Testing different collaboration strategies for expanding events

expanding	**256 (73.33% PR)**				**512 (85.56% PR)**				**1024 (90.00% PR)**			
	top	(%)	total	(%)	top	(%)	total	(%)	top	(%)	total	(%)
strategy												
cooperate	41	75.93	188	35.61	48	70.59	206	33.44	59	60.82	216	33.33
selfish	13	24.07	170	32.20	20	29.41	209	33.93	29	29.90	210	32.41
sensing only	0	0.00	170	32.20	0	0.00	201	32.63	9	9.28	222	34.26
location												
bottom-half	0	0.00	187	35.42	0	0.00	223	36.20	56	57.73	204	31.48
center	35	64.81	236	44.70	33	48.53	215	34.90	29	29.90	240	37.04
top-half	19	35.19	105	19.89	35	51.47	178	28.90	12	12.37	204	31.48
com-type												
push	25	46.30	259	49.05	20	29.41	303	49.19	38	39.18	324	50.00
pull	29	53.70	269	50.95	48	70.59	313	50.81	59	60.82	324	50.00
subset												
10%	28	51.85	265	50.19	11	16.18	333	54.06	25	25.77	345	53.24
100%	26	48.15	263	49.81	57	83.82	283	45.94	72	74.23	303	46.76

---

[1] Our successful run condition is that 95% of the agent population safely exited.

location of the event also matters – events near the exits (bottom-half) delay escaping agents, and reduce the chance that distant agents will learn of the event. We can also see in the fastest cases that a pull strategy outperforms a push strategy; however, they are roughly equal overall. Similarly, an epidemic messaging strategy with a 10% forwarding rate is just as effective as flooding for low density populations; however, a flooding-based strategy leads to (not surprisingly) faster success with larger populations due to more rapid information saturation. Again, when looking at all successful runs, both are similarly effective. Minimizing messages is not necessarily a concern in this setting, but it might be practically relevant that these communication strategies also work if not all agents participate.

Table 2. Testing different collaboration strategies for moving events

moving	**256 (57.64% PR)**				**512 (79.17% PR)**				**1024 (87.64% PR)**			
	top	(%)	total	(%)	top	(%)	total	(%)	top	(%)	total	(%)
strategy												
cooperate	28	65.12	181	43.61	41	67.21	213	37.37	46	66.67	212	33.60
selfish	15	34.88	128	30.84	20	32.79	189	33.16	18	26.09	206	32.65
sensing only	0	0.00	106	25.54	0	0.00	168	29.47	5	7.25	213	33.76
location												
bottom-half	3	6.98	161	38.80	2	3.28	210	36.84	36	52.17	195	30.90
center	19	44.19	156	37.59	22	36.07	178	31.23	31	44.93	230	36.45
top-half	21	48.84	98	23.61	37	60.66	182	31.93	2	2.90	206	32.65
com-type												
push	16	37.21	195	46.99	21	34.43	270	47.37	29	42.03	312	49.45
pull	27	62.79	220	53.01	40	65.57	300	52.63	40	57.97	319	50.55
subset												
10%	21	48.84	206	49.64	8	13.11	302	52.98	8	11.59	323	51.19
100%	22	51.16	209	50.36	53	86.89	268	47.02	61	88.41	308	48.81

Table 2 shows the results for *moving* events, which can obstruct exits and disturb agents' exit routes. As we can see in Table 2 (similar to Table 1), low density populations are less likely to achieve high overall success compared to higher agent densities.

4.3 Evaluation of Different Decentralized Coordination Strategies

Figures 3–5 capture the numbers of agents informed over time based on different collaboration strategies, types of events and agent populations (different color lines for each density, solid for push and dotted for pull). Figure 3 shows a stationary event located in the center of the space. As can be seen, a cooperative strategy leads to (for the highest agent density, in purple) nearly 70% of agents being informed quickly and then exiting the space quickly (around 90 ticks). Under the selfish and sensing only strategies, agents remain in the space until nearly everyone is informed or has encountered the event first hand. Figure 4 depicts an event centered in the space that expands over time. Due to the dynamic changes of the event, agents are informed quickly, but spend more time in the space due to the need to adapt and 'replan' their exit route, slowing their exit process. Figure 5 presents the results of a moving event, which show similar results compared to an expanding event.

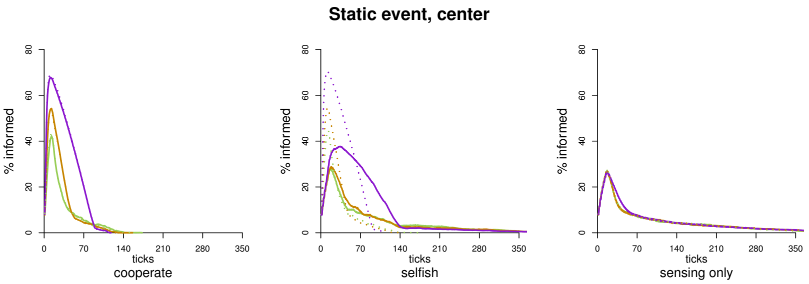


Fig. 3. Informed agents over time with a stationary event in the center

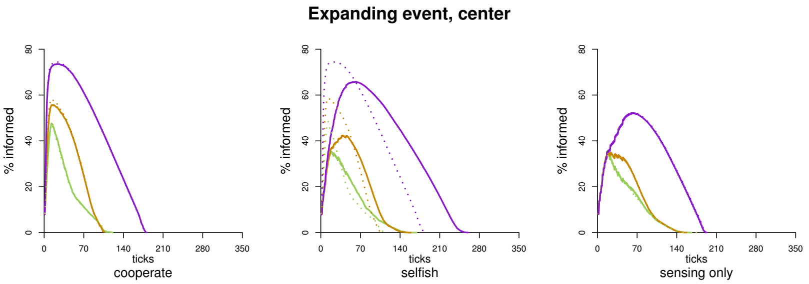


Fig. 4. Informed agents over time with an expanding event in the center

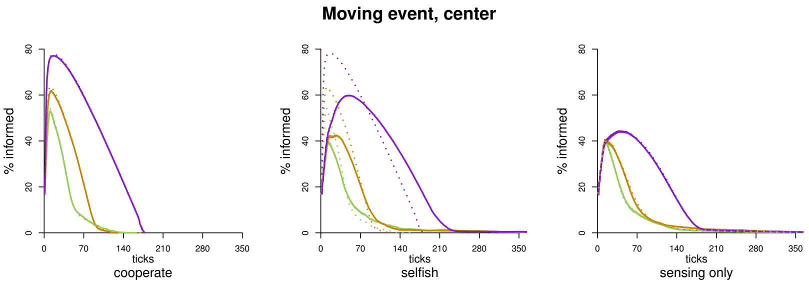


Fig. 5. Informed agents over time with a moving event in the center

In summary, for nearly all scenarios examined, the cooperative agent strategies far outperform their sensing only (uncooperative) or selfish counterparts. Under cooperative cases, initial event-related information dissemination occurs more quickly, and across a larger percentage of the total population than under uncooperative and selfish models, and as a result, a higher percentage of the total agent population was able to successfully exit, more quickly. The goodness of communication strategies (push or pull; epidemic or broadcast routing protocols) varies slightly depending on the specific event type and initial location; but there is no clear or consistently better selection. Results do however indicate that event type and location (relative to exits or goals) plays a significant role in the emergent agent behaviors over time.

5 Related Work

Today, several simulation tools exist that facilitate the investigation of various aspects of social agents collaborating in spatio-temporal environments [1, 2, 4, 11]. While many of these tools have gained widespread publicity, there is currently no single simulation environment that allows practical investigation of all the components (technical, social, and environmental) relevant to our proposed application.

The idea of decentralized ad-hoc collaboration has been successfully established in wireless sensor networks and especially in geosensor networks, in which the concept of location, local events and node proximity to spatial events is poignant. Ad-hoc collaboration has also been used to aggregate local sensor information to form knowledge about global event such as establishing and tracking event boundaries [8, 9]. In *mobile* geosensor networks, sensor nodes use ad-hoc collaboration with nodes they encounter in spatial proximity to pass on information about e.g. hazardous road conditions [14], exchange information about potential rideshares in intelligent transportation networks [16], or collaborate on capturing currents in ocean sensor networks [15].

While [17, 18] explore ad-hoc collaborative decision making in spatio-temporal environments; this work focuses on complex collaborative tasks such as toxic spill clean-up and agents with varying abilities. Our framework could be useful for exploring collaboration strategies for more complex tasks in emergency situation such as rescuing victims or directing crowds through a space that is unknown to most participants.

[20] explores an idea that is similar to ours as presented in this paper. However, this work focuses more on the representation and sharing of partial spatial knowledge and creating ad-hoc local maps of a graph/map structured outdoor environment using only a broadcast strategy, while our work investigates several different communication protocols (push vs. pull, and broadcast vs. epidemic) and also explores various types of events (e.g. moving events). We also propose the *LocalAlert* simulation environment as the basis for more exploration under this research question. Overall, both approaches come to similar conclusions; mainly that ad-hoc collaboration enables a better outcome in emergency situations.

6 Conclusions

In this paper, we investigated the potential of smartphone based ad-hoc collaboration in emergency situations. We presented the *LocalAlert* simulation framework, designed to simulate various ad-hoc collaboration protocols for agents dynamically reacting to spatio-temporal events. We tested agents acting alone (sensing only), selfishly, and under a fully cooperatively behavior model, and our results indicate that such an application is indeed valuable. Under cooperative models, information dissemination occurred most quickly over the largest percentage of the population, and as a result, a greater percentage of the total population was able to successfully exit in less total time. This paper serves as a first exploratory analysis of several possible and likely communication and coordination strategies.

In the future, the *LocalAlert* framework will be used to perform a much more in-depth analysis. We also make the *LocalAlert* framework available as open source code so that it is available to the community for continued development of new modules and to introduce other options, such as additional communication models, agent social behaviors, or spatial layouts. There are still many open, interesting research questions related to this work, which need to be addressed in other research areas of GIScience. For example, which human user interface is most appropriate for such an application? What is the best way to represent imprecise spatial knowledge and support automatic reasoning about it? How can we aggregate imprecise spatial knowledge from different sources automatically? The authors of this paper plan to continue exploring such interdisciplinary questions, and hope that this work serves to encourage other GIScience researchers to do the same, so that a real-world implementation of the *LocalAlert* framework may one day exist.

References

1. Buzing, P., Eiben, A., Schut, M.: Evolving agent societies with vuscape. In: Banzhaf, W., Ziegler, J., Christaller, T., Dittrich, P., Kim, J. (eds.) Advances in Artificial Life, Lecture Notes in Computer Science, vol. 2801, pp. 434–441. Springer Berlin / Heidelberg (2003)
2. CASOS: Construct. `http://www.casos.cs.cmu.edu/projects/construct/`
3. City of Cupertino: Ready 95014 iPhone/iPad App, `http://www.cupertino.org/index.aspx?page=87`
4. Epstein, J., Axtell, R.: Growing Artificial Societies: Social Science from the Bottom Up. The MIT Press (1996)
5. Estrin, D., Govindan, R., Heidemann, J., Kumar, S.: Next century challenges: scalable coordination in sensor networks. In: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking. pp. 263–270. MobiCom '99, ACM, New York, NY, USA (1999)
6. European Telecommunications Standards Institute EMTEL: Analysis of the Short Message Service (SMS) and Cell Broadcast Service (CBS) for Emergency Messaging applications. Tech. Rep. 102 444 V1.1.1, ETSI, February (2006)
7. Feeney, L.: A Taxonomy for Routing Protocols in Mobile Ad Hoc Networks. SICS Research Report (1999)
8. Jiang, J., Worboys, M., Nittel, S.: Qualitative change detection using sensor networks based on connectivity information. GeoInformatica 15, 305–328 (2011), `http://dx.doi.org/10.1007/s10707-009-0097-0`, 10.1007/s10707-009-0097-0

9. Jin, G., Nittel, S.: Efficient tracking of 2d objects with spatiotemporal properties in wireless sensor networks. Distributed and Parallel Databases 29, 3–30 (2011), `http://dx.doi.org/10.1007/s10619-010-7075-2`, 10.1007/s10619-010-7075-2
10. Mercury News: City of Cupertino launches emergency app for iPhone, iPad, `http://iphoneapps-review.net/city-of-cupertino-launches-emergency-app-for-iphone-ipad/`
11. Moon, I.C., Carley, K.M.: Self-organizing social and spatial networks under what-if scenarios. In: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems. pp. 252:1–252:8. AAMAS '07, ACM, New York, NY, USA (2007), `http://doi.acm.org/10.1145/1329125.1329430`
12. Nielsen: NielsenWire, Nieslen Blog. `http://blog.nielsen.com/nielsenwire/?p=28237`
13. Nittel, S.: A survey of geosensor networks: Advances in dynamic environmental monitoring. Sensors 9(7), 5664–5678 (2009), `http://www.mdpi.com/1424-8220/9/7/5664`
14. Nittel, S., Duckham, M., Kulik, L.: Information dissemination in mobile ad-hoc geosensor networks. In: Egenhofer, M.J., Freksa, C., Miller, H.J. (eds.) Geographic Information Science, Lecture Notes in Computer Science, vol. 3234, pp. 206–222. Springer Berlin / Heidelberg (2004)
15. Nittel, S., Trigoni, N., Ferentinos, K., Neville, F., Nural, A., Pettigrew, N.: A drift-tolerant model for data management in ocean sensor networks. In: Proceedings of the 6th ACM international workshop on Data engineering for wireless and mobile access. pp. 49–58. MobiDE '07, ACM, New York, NY, USA (2007), `http://doi.acm.org/10.1145/1254850.1254860`
16. Nittel, S., Winter, S., Nural, A., Cao, T.: Shared ride trip planning with geosensor networks. In: Miller, H.J. (ed.) Societies and Cities in the Age of Instant Access, GeoJournal Library, vol. 88, pp. 179–194. Springer Netherlands (2007)
17. Raubal, M., Winter, S.: A spatio-temporal model towards ad-hoc collaborative decision-making. In: Painho, M., Santos, M.Y., Pundt, H. (eds.) Geospatial Thinking, Lecture Notes in Geoinformation and Cartography, vol. 0, pp. 279–297. Springer Berlin Heidelberg (2010)
18. Raubal, M., Winter, S., Dorr, C.: Decentralized time geography for ad-hoc collaborative planning. In: Hornsby, K., Claramunt, C., Denis, M., Ligozat, G. (eds.) Spatial Information Theory, Lecture Notes in Computer Science, vol. 5756, pp. 436–452. Springer Berlin / Heidelberg (2009)
19. Wilensky, U.: NetLogo. `http://ccl.northwestern.edu/netlogo`. Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL (1999)
20. Winter, S., Richter, K.F., Shi, M., Gan, H.S.: Get me out of here: collaborative evacuation based on local knowledge. In: Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness. pp. 35–42. ISA '11, ACM, New York, NY, USA (2011), `http://doi.acm.org/10.1145/2077357.2077365`
21. ZigBee Alliance: ZigBee. `http://zigbee.org/`
22. ZigBee Alliance: ZigBee Telecom. `http://zigbee.org/Standards/ZigbeeTelecomServices/Overview.aspx`