Web-based collaborative document writing for emergency management

Adrian Shatte
College of Business, Law and Governance (Info. Tech.)
James Cook University
Townsville, QLD, Australia

Jason Holdsworth
College of Business, Law and Governance (Info. Tech.)
James Cook University
Cairns, QLD, Australia

Ickjai Lee
College of Business, Law and Governance (Info. Tech.)
James Cook University
Cairns, QLD, Australia

Abstract

This paper describes the progress of our research into collaborative document writing for emergency management. Our focus is specifically on the use of web technologies to improve communication and shared knowledge in emergency situations. We present a prototype system built using Differential Synchronisation with flexible locking and user attribution features. Based on the past shortcomings of technologies and documents such as situation reports, we consider the potential benefits of a web-based system with flexible locking and user attribution for collaborative situation report writing.

1. Introduction

Collaboration is a fundamental technique used in the creation and development of work artefacts over time [1]. For many tasks, the participation of multiple users provides skill diversity, sharing of the workload, and keeps interested stakeholders up to date with the progress and direction of a project [1].

Emergency management is a complex task that requires knowledge sharing between many separate entities including law enforcement, government, and the public [2]. In an emergency situation, communication between these agencies is important to ensure that accurate information is relayed to stakeholders in a timely manner. One of the important collaborative tasks conducted by agencies during an emergency is report writing [3]. Documents including Situation Reports (sitrep) and Incident Reports may require input from several expert organisations. Further, the content of these documents is constantly changing based on the progression of events and availability of information. This makes the accuracy and maintenance of such documents challenging during disaster periods.

Reviews of past disasters have noted challenges and shortcomings in the use of sitreps for disaster management. For example, a recent review of the response to the Christchurch Earthquake found that sitreps were ineffective due to poor coordination of incoming information, changing requirements, and conflicting priorities between departments [4]. The report also noted that the response relied on personnel from a single company in a single location, which promoted cohesion but increased vulnerability [4]. Real-time, online collaborative editing systems with diverse features could strengthen the response to disasters, primarily in the upkeep and accuracy of sitreps and promoting input from the most relevant stakeholders despite their geographic locations [5].

Two features that could have benefits for collaborative emergency management systems are flexible locking and user attribution. Flexible locking can ensure the integrity of shared data by providing fine-grained control over edit privileges [6]. Such a locking system should be optional (users can choose to lock sections of shared text) and dynamic (the locks adjust according to the context of other edits). User attribution refers to techniques for increasing awareness in a shared workspace by attributing changes to a user [7]. If this information is displayed visually it can improve communication and result in better collaboration [8].

Differential Synchronisation (diffsync) is a synchronisation technique used to converge shared data between multiple collaborators [9]. With the recent addition of flexible locking and user attribution features [10], diffsync can provide many benefits for collaborative document writing in emergency situations. For example, expert authors can claim exclusive editing privileges over any section of the document to prevent conflicting or erroneous edits. Further, user attribution and visual representation of authorship provide readers with immediate knowledge about the agency that contributed that information.

In this paper, we consider the benefits of dynamic locking and user attribution in diffsync to develop a real-time collaborative editing system for emergency management. This paper begins with a background in
collaborative communication for emergency management and real-time collaborative editing tools with a specific focus on flexible locking and user attribution. Next, we describe our proposed framework for developing systems using diffsync with flexible locking and user attribution. Using this framework we developed a web-based prototype for collaborative emergency document writing, and we demonstrate its applicability to an emergency management scenario in Section 4. Finally, we discuss the implications of our study and consider future work in the area.

2. Preliminaries

2.1. Collaborative communication in emergency management

There are four phases of the emergency management process: mitigation, preparedness, response, and recovery. Mitigation refers to the pre-disaster identification of risks and steps taken to reduce potential negative effects of a disaster on property and people. Preparedness refers to warning systems and other processes engaged before a disaster to ensure an adequate response by emergency managers and the public once a disaster actually occurs. Response refers to any action taken immediately prior, during, and after a foretold disaster which assists in reducing human and property loss. Finally, recovery refers to the process of returning the affected population back to a “normal” state after a disaster event [11]–[13].

Information and Communication Technologies (ICT) can be employed to assist emergency management in all four of these phases. The main benefits of ICT for emergency management are: 1) diverse communication channels can be utilised for effective warnings and notifications; 2) information from diverse sources can be easily integrated into a single system; 3) disaster relief efforts can be centrally coordinated; 4) systems can encourage both public and institutional input; and 5) damage caused and other effects of a disaster can be analysed to inform recovery [14].

With increased availability of the Internet and new ICT technologies, several software systems for emergency management have been developed in the past few years. Table 1 displays some of the systems currently being used in emergency management and the main features of these systems.

<table>
<thead>
<tr>
<th>Name</th>
<th>Main Features</th>
<th>Reference/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebEOC</td>
<td>Event reporting, situational awareness, resource</td>
<td>[15]</td>
</tr>
<tr>
<td>ERIC</td>
<td>Integrates data from several agencies into single web-based interface, historical snapshots and automated sitreps.</td>
<td>[16]</td>
</tr>
<tr>
<td>ESA</td>
<td>Detects unusual behaviour on Twitter to quickly alert when a disaster event is being broadcast.</td>
<td>[17], [18]</td>
</tr>
</tbody>
</table>

The effectiveness and reliability of such systems relies on the underlying technologies supporting real-time and responsive synchronisation. The challenge of heterogeneous agencies involved in emergency management can be reduced by systems that provide real-time status of activities and conflicts [19], [20]. Section 2.2 discusses real-time collaborative editing tools that are useful for building such systems.

2.2. Real-time collaborative editing tools

Several techniques have been proposed to maintain consistency in collaborative software, including pessimistic algorithms [21], shared content management systems and wikis [22], state-based algorithms [9], and edit-based algorithms [23]. These various techniques handle the challenges of consistency maintenance and conflict management in different ways, and some support other features that are beneficial for collaborative editing tasks including user attribution and flexible locking.

Operational Transformation (OT) is a popular edit-based technique for synchronising shared data in collaborative applications [24]. OT works by propagating changes between collaborators as operations (e.g. insert text, delete text, and so on) and transforming those operations at remote sites to repair inconsistencies [24]. However, edit-based techniques like OT can prove problematic to implement due to the rich nature of modern collaborative interfaces requiring complex operations [9].

Diffsync is a client-server approach to collaboration that works by comparing document states and merging changes. Diffsync has many benefits that make it ideal for use in collaborative writing for emergency management, including scalability, convergence, and fault-tolerance in high latency or unreliable network environments [9]. Further, the synchronisation mechanism of diffsync is independent of the user interface meaning it can be appended to existing single-user applications [25]. With the recent addition
of features such as flexible locking and user attribution [10], diffsync is a suitable candidate for developing collaborative writing applications for emergency management.

2.3. Flexible locking in collaborative editing

A simple example of document locking can be encountered when two users attempt to edit a Microsoft Word document that is shared on a local server – Microsoft Word locks the document for editing to the first user who accessed it, and any subsequent users who open the document are provided with read-only access until the first user is finished with the document and closes it. While this approach may achieve consistency, it inhibits usability and does not scale well to real-time collaboration on the web. As such, researchers have developed algorithms to support synchronous editing with optional and dynamic locking which is essential in collaborative writing for emergency management.

The majority of the work on optional and dynamic locking for collaborative algorithms comes from the field of edit-based algorithms, specifically the OT technique [24]. For example, Sun [26] proposes a scheme that is optional, non-blocking, responsive, and allows for multiple locks to be used simultaneously inside a single document editing session. Similarly, Xu et al. [27] propose customisable and dynamic locking in which the locking mechanism can be adjusted with separate locking policies, and the system can pre-lock sections automatically based on the enforced policies and recent activity in the shared workspace. More recently, applications for collaborative programming have been developed using diffsync with flexible locking [10], [28].

Flexible locking is a feature that could have benefits for real-time collaborative document editing systems for emergency management. Optional locking allows a collaborator to maintain the integrity of a section of text to prevent erroneous or mutually exclusive edits from occurring. This is important in emergency situations that rely on accurate information. Further, optional locks can be used to delegate work by allowing different staff or departments to claim exclusive ownership and responsibility for a section of the document.

2.4. User attribution in collaborative editing

Maintaining awareness of user intention, actions, and results is crucial in determining the success of a collaborative task [29], and is of great importance in emergency management [30]. Workspace awareness involves providing knowledge of who is present in the shared workspace, where they are working, what they are changing, when those changes are made, how things are changed, and why the changes are made [1], [31]. Mechanisms for communicating this information to the user help to coordinate activity, simplify communication, provide relevant assistance, and manage the movement between individual work and shared work [31].

There are many different methods used to provide user attribution in shared workspaces. These methods include visual cues [1], [7], eye tracking [32], [33], and mouse and cursor tracking [34]. These features can be associated with a user profile to indicate additional user attribution. It is believed that user attribution features can assist in supporting real-time collaboration for emergency management document writing. In emergency situations, the reliability and accuracy of information is important, so knowledge of an author’s identity and affiliations can indicate that a source is trustworthy. Further, as the number of collaborators increases in a shared emergency document, it can become confusing as to where information is coming from.

3. Proposed framework

Our framework is built on diffsync for synchronisation [9]. We also integrate user attribution and flexible locking features into diffsync for use in our system. In this section, we describe a prototype system for collaborative document writing in emergency management. This includes a discussion on the research methods guiding our process, the software tools we used, and the main features of our prototype system.

3.1. Research methods

Our research and implementation are guided by the Design Science approach [35]. As the development of our system and its underlying technology is ongoing, the results of this study will inform future work and development.

3.2. Software used in our prototype

Our prototype is developed using several types of software, including Ace editor, Node.js, and Google Maps. These technologies are briefly described below in terms of their interoperability and benefits for real-time editing.

3.2.1. Ace editor. Ace editor is a web-based code editor that is compatible with JavaScript applications.
Ace provides many features common to native Integrated Development Environments (IDEs) including syntax highlighting, indentation, line numbers, and custom themes for different programming languages. Ace is capable of supporting small and large documents, which makes it useful for application scalability. We selected Ace editor to exploit its highlighting functionality for user attribution, and also for its compatibility with Node.js.

3.2.2. Node.js. Node.js is a highly scalable, server-side JavaScript environment based on Google’s V8 runtime engine [36]. It is commonly used to develop real-time applications (such as web servers and networking tools) due to its single-threaded, asynchronous nature. Node.js is also non-blocking. Furthermore, Node.js is supported by many popular cloud providers (e.g. Amazon Web Services [37] and Google App Engine [38]) which means our application can be deployed using the Platform as a Service (PaaS) model. All of these features make useful for developing real-time, collaborative text-editing systems.

3.2.3. Google Maps API. The Google Maps JavaScript API provides public map features for web-based applications. Key features include road map and satellite imagery, integrated data from diverse sources (e.g. business information, landmarks, etc.), customisation, and advanced visualisation features (e.g. heat maps and symbols). We selected Google Maps for our prototype to demonstrate a heat map representing incoming incident reports. As the Google Maps API is written in JavaScript it is compatible with our other selected technologies.

3.3. Diffsync

For collaborative editing features, we use the Guaranteed Delivery implementation of diffsync [9]. Diffsync is a robust algorithm for document synchronisation with open source implementations written for common programming languages, including JavaScript, which means that it is compatible with the other tools used in our prototype. We selected diffsync for our prototype due to its ease of implementation, high responsiveness, and high concurrency rate [9].

A number of characteristics make diffsync appropriate for real-time collaborative applications such as: 1) nearly identical code on both the client-side and the server-side; 2) a state-based approach that naturally provides an edit history; 3) asynchronous updates and scalability; 4) fault tolerance on unreliable or high-latency networks; 5) all clients are guaranteed to converge to the same document; and 6) compatibility with various document types.

Finally, diffsync is designed so that it can be appended to existing applications, thus introducing real-time collaboration to previously static systems [9]. That is, existing single-user applications can be provided with powerful collaborative features and adapted to suit a desired type of collaboration in an efficient manner.

Figure 1 illustrates the symmetrical nature of the synchronisation technique. The client sends edits to the server in the form of diffs, and the server patches those edits onto the shared text. The process is repeated in the opposite direction from the server back to the client. Figure 2 demonstrates the topology of diffsync running with several clients. Each client continuously runs the synchronisation loop with the server and consistency is maintained.

![Figure 1: Diffsync runs a continuous synchronisation loop between the client and server.](image)

While diffsync can provide real-time synchronisation, its design also makes it robust to unreliable network conditions that may exist in an emergency situation. For example, its asynchronous nature eliminates the blocking of user input while waiting for a network response. The “Guaranteed Delivery” implementation described by Fraser [9] incorporates additional features to maintain convergence when packets are lost, duplicated, or corrupted. Further, alternative synchronisation mechanisms such as push-based synchronisation have been proposed to reduce dependency on a stable connection [25].

Using the software discussed in this section, we developed the emergency management prototype for collaborative document writing described in Section 3.4.

3.4. Emergency management prototype

We have developed a web-based prototype for collaborative document writing with a specific focus on emergency management. The application runs in a browser and has many features including flexible locking, user attribution, document change history, real-time chat, and a heat map display. Figure 2 shows
the interface of our system. Each feature is explained in more detail in the following sections.

3.4.1. Flexible locking. Our web-based prototype includes features for flexible locking. Users can select any region of text within the shared workspace and request a lock. This lock request is sent to the server on the next synchronisation loop. The server maintains a record of current locks, and upon receiving a new lock request, it checks for conflicts and returns a response to the client on the outgoing cycle. If the lock request is valid, the client is allowed to continue editing uninhibited. Otherwise, the user is banned from further edits in the locked region.

Locks are visually represented to the user by a solid colour highlighting the locked region of text. Each client has a unique colour that represents their identity in the shared workspace. This provides awareness that a region of text is locked and allows quick identification of the owner of that lock. These visuals may affect readability of the text so there is also an option to toggle them off. This hides the colour highlighting but does not affect the integrity of the locks.

In the scope of collaborative writing for emergency management, this locking feature allows experts or authority figures in a situation to claim exclusive ownership over a section of the document. During an emergency situation, information can change quickly. While a section of the document is locked its contents can be verified for accuracy and integrity without the risk of changes being made. Further, erroneous or conflicting edits can also be prevent from occurring during the period of the lock.

3.4.2. User attribution. Our web-based prototype also includes several features used for author and user attribution. To begin the server determines the contributions of each user and maintains a record. This information is updated and propagated to the clients on each synchronisation loop. Using this data the clients render several visual cues.

First, each section of text is underlined with a solid colour to denote authorship. The identity and colour of each client is displayed in the top right corner of the text editing window as a quick reference point. This technique allows for quick and simple identification of the entity that contributed any section of the text. A tooltip is also displayed whenever the mouse is hovered over text, displaying the username of the author. These features can be toggled off at the client-side for read-only mode.
It should be noted that the colours used to denote authorship are also used to denote locking, as mentioned in Section 3.4.1. However, the locking display is semi-transparent whereas the attribution display is fully opaque, which means that a user can easily distinguish between author and lock owner when they have different identities.

These features benefit collaborative document writing for emergency management by increasing situational awareness. Emergency personnel working in the shared document will know who is currently making changes in the document, the location of those changes, and also the content of those changes. If content is attributed to a specific user communication will be improved, as users are held accountable for their edits and can be called upon for clarification if necessary.

3.4.3. History. A related feature to user attribution is history. The server maintains a record of the unique versions of a document (e.g., any time text is added or removed) and the user attribution status of the document at that time. The client can request this history which is displayed on a separate page. The history feature allows users to identify the progression of a document over the duration of a disaster period using a slider (see Figure 3). Secondary benefits are that the user can recall contributions that may have since been deleted, and also determine which agencies contributed the most over time.

3.4.4. Chat. Real-time chat is facilitated by the server to provide additional awareness in the collaborative workspace. The intention of users is not always clear when working on a shared document, so informal chat can allow users to justify edits and also plan future changes before committing them to the live document. This is important in an emergency situation when timing and accuracy of information are critical. We further incorporate the user attribution features into the chat window by displaying each message in a coloured border (see Figure 4).

3.4.5. Heat Map. The final feature in our prototype is a dynamic heat map, displayed in the top right corner of the workspace. This map displays additional information about the relevant emergency situation (see Figure 5). In our case, we have simulated the location of incident reports from Twitter. Public information from social media is important in a real-time disaster management situation but it needs to be verified for accuracy.

4. Case study - Sitreps

To consider the usefulness and applicability of our collaborative prototype in document writing for emergency management, we consider the sitrep document. In this section of the paper, we describe the details of sitreps, potential weaknesses in the process of writing sitreps, and discuss benefits of our prototype at overcoming these challenges with specific focus on flexible locking and user attribution.

4.1. Sitrep (Situation Report)

An important collaborative document used by agencies in an emergency situation is the sitrep. A sitrep is a document that records an ongoing description of all events occurring during an incident.
(e.g. emergency, disaster, and so on). The content of a sitrep must be both accurate and timely, as these details are passed on to adjacent agencies or higher headquarters and Emergency Operations Centres. The information contained in the report is used to make critical decisions during an emergency [39].

The sitrep can appear in many formats. The Australian Government suggests that a sitrep contains a report number, date and time, type of incident, location of incident, contact details of the relevant incident management centre, casualties (including dead, injured, evacuated, and homeless), general details about the situation and damage, actions in progress, assistance required, future intentions, and the overall prognosis of the situation [39]. Other guidelines state that a sitrep should contain factual information only, be brief, and should be specific to a single functional area (e.g. roads). Sitreps can also include graphics such as maps.

While sitreps can be useful for providing brief but informative facts about an emergency incident, there can be challenges in using them effectively. Several reports have noted that lack of effective coordination, poor communication and confusion over situation awareness result in ineffective sitreps. For example, a report on the response to the 2011 Christchurch earthquake noted that sitreps were not always shared with the appropriate parties, resulting in important information being unavailable to assist with decision making [4]. In addition, many parties wasted time compiling duplicate reports and responses for different levels of government when that information was already available in previous versions of sitreps [4].

In addition, supporting technology can also negatively impact information sharing in an emergency situation. The Royal Commission into the 2009 Victoria Bushfires concluded that key agencies used different technologies which caused access difficulties for staff at incident control centres. This inhibited the propagation of important information including sitreps, notifications, warnings, and maps [38].

In light of the challenges and shortcomings of sitreps and ICT in previous disaster situations [4], [40], we considered how our use of diffsync with flexible locking and user attribution may provide benefits to collaborative sitrep writing.

4.2. Benefits of flexible locking and user attribution for collaborative sitreps

One of the key benefits of collaborative editing systems is that all collaborators share a single workspace. In an emergency situation it is crucial that all parties have access to the same information so that decision-making can be coordinated. As diffsync is naturally convergent and provides fault tolerance for intermittent or high latency network environments, divergent copies of the document are avoided. It also means that the current version of the document displayed in the system is always the most up-to-date copy.

Coordination is also improved in a single workspace, as all parties must access the sitrep in a single location. The chat feature of our system increases coordination and awareness as users can justify or explain the reasoning behind edits that may not be clear (see Figure 6). Further, because the document is being edited in real-time, other users can pick up on mistakes or ambiguous edits and fix them immediately.

The user attribution features of our prototype ensure that users are responsible for their own contributions. This may have effects on the quality and accuracy of contributions in an emergency situation as organisations and individuals are held accountable for their work. Awareness is also increased as users will know who is working in the document, what changes they are making, and where those changes are being made. This is important in providing context to the content, as users can identify the accuracy and reliability of information based on its source (see Figure 6).

Flexible locking means that users can take exclusive ownership over any section of the document as necessary. When a section of text is locked, only the owner of that lock can make any changes. This can be important when specialised expertise is required for reporting on some aspect of a disaster. The expert user in that topic can lock the region of text to prevent conflicting, erroneous, or mutually exclusive edits from being made while data is verified. In addition, a user with higher authority can lock other users from editing the document while the content is checked or revised (see Figure 6).

Finally, the history feature of our technique may improve situation awareness. In an emergency situation, the status of an emergency and incoming information can change quickly. The sitrep will be continuously updated to reflect the current situation. Users can review past changes and who made those changes to better assess how the emergency situation and sitrep has changed over time. It can also enforce a higher level of accountability for changes that have been made in the document, as other users can review the actions of any user.

4.3. Challenges of network connectivity

It should be noted that our current system has a heavy dependency on a stable network connection.
While diffsync is robust to network faults such as packet loss or data corruption [9], complete loss of Internet connection would render the system not operational. For this reason it is important to consider alternative network solutions that could support our system in an emergency situation. Nilsson and Stølen distinguish four main network solutions for emergency response: 1) wireless ad hoc networks; 2) cellular networks; 3) special emergency networks; and 4) router-based networks that are deployed for the emergency operation [41]. They conclude that wireless ad hoc networks are well-suited for emergency response as they are quick to deploy, provide local communication within the disaster area, and can be used alongside an Internet connection (if it is available) [41].

5. Final remarks

In this paper we considered the benefits of a web-based, collaborative editing system for document writing in emergency situations. Specifically, we focused on the benefits of flexible locking and user attribution features in improving awareness and accuracy for the writing of sitreps. We cited some of the past challenges with ICT and sitreps for emergency management and consider how our technique could overcome these challenges and better support collaboration.

The prototype and discussions presented in this paper indicate the need for further work. In terms of technical contributions our system does not support rich text editing or document storage for different sitreps. This means that each department or organisation would still need to log in to a separate instance of our system to access different sitreps. Better mechanisms for moving between different sitreps should be introduced as well as integration with external information (e.g. Bureau of Meteorology, Department of Transport, and so on).

Another consideration for future work is support for mobile technologies. Currently our prototype supports users on a desktop or notebook computer. However there could be users on the scene of an emergency who have a better understanding of the situation. In the next iteration of development we will consider methods to allow mobile users to contribute to the writing of reports.

There is also room for improvements to be made to the document history feature. Our current implementation saves a version of the document any time a change has been made (either text added or deleted). This can result in a large history being recorded over a short amount of time. In an emergency situation, users will need to have methods to navigate quickly through the history of the document to find relevant information. Thus, we will explore methods to categorise and better record history of a document.

Finally, we noted the dependency of our system on a stable Internet connection. While alternative network solutions to manage this dependency were noted in Section 4.3, further investigation is necessary to ensure that synchronisation can be maintained when conventional networks fail.

To conclude, our research in this area has indicated that collaborative writing with flexible locking and
user attribution may benefit document writing for emergency management. We intend to build upon this research in the future to investigate the usefulness of these technologies in other areas of emergency management. We also plan to incorporate additional features and improve on the technical aspects of the system.

6. References


