User Experiences of Incident Reporting Software in Fire Services: An Integrative Review and Meta-Analysis

Aimee Kendall Roundtree(✉)

Texas State University, San Marcos, TX 78666, USA
akr@txstate.edu

Abstract. This integrative review gathers data from published articles and user feedback for a meta-analysis of the common problems, use contexts, and recommendations. The project supplements primary interviews with secondary data from prior studies and reports, as well as online feedback and reviews. This approach helps validate user experience findings for rarely-tested products, and it helps to confirm and identify user affordances and system pain points. Findings suggest that poor visibility of system status, lack of match between system and real-world use, and opaque help and documentation are common barriers. NFIRS software programs also do not anticipate the cultural idiosyncrasies endemic to fire services (such as apprenticeship learning) that, if addressed, could help software users better recognize, diagnose and recover from decision-making errors. Firefighters request more functionality, more help to find pertinent codes, and differentiating between nondescript codes. Recommendations for improving the quality of software programs for incident reporting in fire services include improving customization features, providing templates and content guides, and improving the glossaries of common acronyms. Help systems should address the diverse backgrounds and levels of education that comprise fire services.

Keywords: Human factors · Fire services · Integrative review

1 Introduction

This integrative usability review gathers data from published articles and from user feedback to perform a meta-analysis of the common problems, use contexts, and recommendations. The project supplements contextual and individual interviews with data from prior studies and reports, as well as online feedback and reviews. This approach contributes to usability research by enlisting integrative review strategies to find and synthesize data on incident report systems in fire services. Integrative reviews are a type of review more rigorous than standard, narrative reviews insofar as they analyze experimental and non-experimental findings by combining evidence of multiple primary studies in order to derive generalizable findings from consensus [1–4]. Integrative reviews also enable merging qualitative and quantitative findings by reflecting on the interaction between ratings or numbers and unstructured feedback [5, 6]. By using this mixed methods approach, this project aggregates feedback from

F. F.-H. Nah and K. Siau (Eds.): HCII 2019, LNCS 11589, pp. 367–378, 2019.
https://doi.org/10.1007/978-3-030-22338-0_30

usability tests and user public comments for the purpose of formulating general heuristics to help fire services administrators make more informed decisions about the software they choose, rather than singling out a particular problem or issue with any particular vendor's product.

This study also makes a contribution by sharing a comprehensive analysis of feedback regarding incident report writing for firefighters and first responders. While the software in the fire services sector most likely undergoes user testing prior to product launch, the proprietary nature of these test results precludes their wide distribution. Considering preexisting and public user feedback about the software might encourage and inspire innovative developers to help solve the issues that persist in incident report writing software.

The objective of this integrative review is to aggregate and synthesize publicly-accessible user feedback pertaining to incident report software in order to determine the common pain points and problems for designers and developers to resolve. The research question that this article will answer is as follows: What problems with design and help documentation persist across incident reporting software products for firefighters and first responders? Findings reveal that interface design may impact decision making and report quality. In the case of fire services, interface design that facilitates decision making is essential.

Background. Software programs for incident reporting in fire services abound. According to the U.S. Fire Administration, there are 110 active National Fire Incident Reporting System (NFIRS) vendors and 86 registered vendors [7, 8]. The usability of these software programs is integral to accurate reporting which, in turn, ensures that fire services have the best data necessary for making decisions about fire risk, resource allocation, education, and outreach in a community. One fact anchors the most important issues for improving the user experience of incident reporting in fire services: The software must simplify the decision-making process because the complexity of incident reporting often antagonizes and confounds data accuracy.

A recent report on NFIRS found that the Federal system of incident reporting codes has grown too complex and unwieldy [9]. NFIRS incident types include over 175 codes to choose from and interpret; the number of codes has more than tripled from the prior to the current versions of NFIRS. Firefighters use the codes inconsistently due to personal preference, educational background differences, idiosyncrasies of training, and information overload. Fire services have difficulty deciding which are the best software programs to buy, but they have a high standard of accountability not to waste taxpayer dollars. Unfortunately, findings from vendors' usability research and testing of their own software programs often remain proprietary, which disadvantages fire departments around the country insofar as they must gather anecdotal information rather than more robust findings to make decisions about which programs to buy.

Prior studies suggest that several factors play a role in incident report writing difficulties, such as unclear reporting standards, insufficient training and quality control processes, complexity of the NFIRS coding schema, time and stress constraints that characterize the nature of incident response, and uncertainty about the importance and uses of incident report data [10–13]. Three articles also reported problems with software databases and user-friendliness [10, 12, 13]. However, two of these articles did

not provide sufficient information about user feedback to integrate into this study; only one of these articles divulged all of the primary data and direct quotes from users, which was integrated into this study. Heuristics derived from this integrative usability review will not only facilitate fire services software selection processes, they will also help designers conceptualize, revise, and authenticate use cases in the fire services as these designers create and improve incident reporting software.

2 Methods

The integrative review was comprised of three sources of content: usability focus group interviews, a review of prior literature, and online review ratings.

2.1 User Focus Group Interviews

Forty-one firefighters from San Marcos and College Station Fire Departments in Texas were interviewed over the course of four weeks about their experiences using two different NFIRS software packages on the NFIRS active vendors list. The focus groups were authorized by the Institutional Review Board at Texas State University. Participants were contacted and recruited through fire chiefs and marshals at San Marcos and College Station Fire Departments. Chiefs and Marshalls sent emails to schedule interviews per shift availability. Participants were asked to fill out a brief background questionnaire. Focus group questions asked participants to share strengths, weaknesses, and suggestions for improvements regarding their current software. Of the 41 participants, 36 were male and 5 were female. Most (n = 28) were between ages 25 and 54. They had served an average of 13.7 years in fire services (Table 1).

Table 1. Focus group demographics

	Category	Number
Age	18–24	2
	25–34	15
	35–44	13
	45–54	8
	55–64	3
Gender	Male	36
	Female	5
Education	High School	2
	Some College	19
	Associate Degree	8
	Bachelor's Degree	9
	Some Grad School	1
	Graduate Degree	2
Years of Service	Average	13.7

2.2 Literature Review

The review of prior literature included English-language reports and peer-reviewed articles published between 2008 and 2018 and cataloged in Google Scholar, Ebsco, IEEE Xplore, and Web of Science. Search terms included the names of active vendors and their respective NFIRS software, as well as the terms user, usability, NFIRS, firefighting, and fire services to delineate studies and research pertaining to user experiences. The abstract and full text of each item from search results were screened for results from usability testing, surveys or other user feedback. Items were eliminated if they contained no user feedback results or if they solely contained hypothetical user models or heuristic evaluations. An Excel spreadsheet was used for data extraction, including the following categories of information from each item: citation, study methods, and results. Findings were combined using content analysis, whereby emerging themes across reports and studies were compiled. Of the 162 articles identified using the keyword searches, seven (7) met inclusion criteria insofar as they reported user feedback rather than overall heuristic evaluations.

2.3 Software Review Ratings

Customer ratings were gathered using Capterra.com. Search terms included the names of active vendors and their respective NFIRS software. Of the 108 active vendors of NFIRS software currently registered with the U.S. Fire Administration, only 24 NFIRS and emergency response programs were included on Capterra.com and received comments for review. Data were analyzed using content analysis, whereby emerging themes across reports and studies were compiled. Text mining tools—Orange.si and RapidMiner—were used to confirm word and phrase frequencies, as well as text clustering. Orange is an open source machine learning and data visualization program for interactive data analysis workflows. RapidMiner is a software platform for data science that includes data prep, machine learning, and predictive model deployment.

2.4 Results Synthesis

In order to synthesize results, all user feedback including negative comments, complaints, and suggestions for improvements were categorized according to the ten usability heuristics for user interface design offered by the Nielsen Norman Group [14]. These heuristics are broad rules of thumb for building user-friendly software products. Definitions of each heuristic are listed below, per the Nielsen Norman Group:

- *System visibility* calls for software to inform users with timely feedback.
- *Match between the system and the real world* calls for the system to follow real-world conventions and use real-world words, phrases, and concepts.
- *User control and freedom* requires that users be permitted to leave an unwanted state without an extended dialogue.
- *Consistency and standards* mean that concepts and actions should carry the same meaning throughout the software.
- *Error prevention* requires that software design prevents problems from occurring.
- *Recognition rather than recall* requires that developers make objects visible and make all instructions retrievable.

- *Flexibility and efficiency of use* allow users to tailor actions.
- *Aesthetics and minimalist design* mean clear relevant info in helpful interface design help.
- *Diagnosing and recovering from errors* dictates that software clearly indicates the problem and solution.
- *Help and documentation* must be concise, easy to search, task-focused, and it must provide steps.

These heuristics served as categories of problems that emerged from the user feedback. They also served as an intuitive schema for combining study results. Only the online ratings from Capterra provided numerical ratings offered by users assessing the overall quality of the software. Therefore, the online review rating section also integrates qualitative and quantitative feedback.

3 Results

3.1 User Focus Group Findings

In 10 transcripts comprised of 7947 lines of exchanges between moderators and participants covering general questions about incident reporting habits, processes, quality control, training, and software, there were 203 suggestions for changes or improvements to the software.

Participants reported several problems with software functioning (n = 25), including insufficient help for quality control, extra keystrokes, too many steps for some functions, and missing features that participants wanted: "You got to put an asterisk or something in it." There were several general statements of discontent about the overall performance of the incident report software (n = 27). Issues included frustration at glitches, general difficulty using the software, and the time-consuming way that the interface was designed: "It is not user friendly in the least little bit." Many of these general problems emerged from comparing the incident report writing software to other software with which they were familiar (n = 37), such as better-designed EMS software: "And the analytics with the [other program] is much better than in [the incident report program], much better." These suggestions point to an error-prone design that failed to prevent problems from happening.

There were problems with interactions between the incident reporting software and the computer-aided dispatch software (n = 16), especially auto-population errors and case sensitivity that complicated the reporting process: "It auto populates a lot of stuff from dispatch." Discrepancies emerged between observations on-site and dispatch information. They also described general limitations with the software's capacity to meet their needs regarding data collection and analytics (n = 8): "[I]t doesn't bring him the other categories for the data collection." Participants also wanted more word processing features included in the software (n = 3) to help them write narratives: "Even the spell check…is very, very, very poor. Terribly poor." Participants did not have the freedom to use the software as their responsibilities required.

There were several requests for better help documentation and software training (n = 20): "The importance of documentation." And "[the help] is too broad."

Participants wished that there were better quality templates to help them write narratives and decide which codes to use (n = 13): "You can have a system…that auto populates a narrative, but they usually aren't very good." System search tools caused problems for participants (n = 6); searches took too long or were not effective finding what participants needed: "You can search it, but it just takes too long." Help and documentation for incident report software are important for participants.

Long lists and drop-down menus (n = 11) made deciding between codes difficult for participants: "There is a long list of options, and when you give somebody like gives us a lot of options, it's difficult to use the right one." There were interface problems (n = 8), including desire for more tabs to make some features more easily accessible and for a more simple login process, as well as a more easy way to work as a team synchronously: "You can put those type of things in there, but not anything that would specify me as the person [who input the info]." Interface design and aesthetics were sometimes unclear to participants and did not aid in usability.

Participants reported dissatisfaction with how the incident reporting software handled errors (n = 5). They wanted better notification and troubleshooting of errors. "Sometimes they'll lose what they typed, and when they have to repeat it because of some error." Participants also felt that incident reporting software used jargon that was sometimes counterintuitive, which, in turn, made it difficult to find what they needed and use the software to capacity: "I wish there was a way that in [the software], I could put plain language in that describes something that happened, and then [the software] give me a whole list of codes that might apply." They wanted plain language options for correcting errors and finding solutions for software problems.

There were requests for more mobile-friendly versions of the software (n = 8), which would help them record incident observations in the field: "You have to be at the fire station to do it." They also wanted more immediate access to newer versions of the software (n = 16). Unfortunately, in the service sector, limited budgets made it difficult to purchase updates: "We had it for 20 years and we're on version seven." Participants wanted the flexibility to use the software in the field and as the job required.

Overall, firefighters reported that technical support for NFIRS software was not easily accessible. They requested more details to help them make decisions about which codes to use. They wanted more detail, such as more code definitions and expert systems to help them make decisions between codes with titles that are very similar or unclear. They wanted easier access to the software, preferring access by tablet or phone over software designed solely for desktops, because their computers were older and limited in number due to constraints on fire services budgets. Firefighters requests more group work features for NFIRS software, such as the ability for two people to work on a report at the same time. They described how some functionality made reporting difficult. In particular, long scrolling menus for reporting personnel, street addresses, and other details made completing reports cumbersome. Search features were limited, which restricted their ability to search for and across incident types, days, and shifts. Firefighters requested more training resources from software developers on NFIRS software. They also wanted more word processing features such as spell check and sentence construction, as well as glossaries for helping decipher and select medical-related acronyms. However, they disliked some of the autofill features of the software because it disincentivized firefighters from writing accurate, event-specific details, and it made it difficult for them to memorize the incident report codes.

3.2 Literature Review Findings

Methodologies varied among the seven studies, including two surveys, two usability tests (with task scenarios, walkthroughs, and think-aloud protocols), a simulation, an interview, and a Delphi group of experts. The findings suggested that mobility of incident report writing software—portability via laptop—helps them record details on-site during an incidence, but mobile phone internet access proved unreliable and difficult to read, given the small screens [15–17] (Table 2).

Table 2. Prior literature findings.

	Sample methods	Findings	Heuristics
Coleman 2010	Survey of 35 departments	Poor GPS accuracy & mobile usability Inaccessibly small screen	Error prevention Minimal design
Peacock and Forney 2008	Surveys of 400 departments	Language clarifications Programming errors with text strings Automated error checking required Better formatting of input options	Match Error prevention Error recovery Minimal design
Romano et al. 2016	10 usability tests	Interfaces difficult to use Colors confusing or not useful Complex forms with too much & useless info Textual message slow tasks	Flexibility Minimal design Error prevention User control
del Olmon Pueblas 2015	6 cognitive walkthroughs	Awkward interface Hard-to-understand conventions	Minimal design Match
Chronaki et al. 2008	Simulation with 20 organizations, 300 volunteers	Problems over hybrid satellite-WiFi. Problems using PDAs Uncertain security and privacy	Match Flexibility Error prevention
Krueger 2010	Interviews with 20 departments	Text input problems Insufficient help & glossaries Sign-in problems Antiquated design	Error prevention Help documentation Minimal design
Averill et al. 2011	Delphi group of 37 fire department reps	Problems with data & format standards Problems merging, manage data sets Interfaces difficult to use, maintain Problems with customization No automatic entry or quality control No peer review or data transfer Make software available on multiple platforms	Consistency and standards Error prevention Flexibility Error diagnosis Match

Even when flexibility of platform choice is provided, incident report writing software is still error-prone. Forms must tolerate appropriate sentence and text string lengths, error checking logic must predict common problems, and layout of input options must promote ease of data entry [18]. For example, according to Peacock and Forney, "[f]ormatting of input options was improved to ease the data entry process. In particular, the structure for entering the fire service apparatus and personnel deployed to the scene was improved" [18, p. 66]. The design must anticipate users' needs and expectations in order to stave errors.

Even when actual time on task values are close to anticipated values, and when users self-reported satisfaction with the interface, first responders may not use the interface widely because they might be more accustomed to using affordances or other traditional methods, such as sharing info face-to-face or over the phone [19, 20]. Matching design to accommodate real-world use is important. Users also found color schemes used by interfaces confusing or unclear [19, 20]. The user interface for software must be intuitive and easy-to-use; able to auto-fill objective information from devices and instruments; designed to calculate time estimates needed to complete data entry on the front end; capable of peer review, ease of configuration, customization and maintenance, available for multiple platforms; accommodating of multiple data standards, types, systems and locations; ready to handle different types of data query and archiving; and able to check for report completeness, standards and quality [21]. As Averill et al. describe, overall, "[t]he user interface for software needs to be intuitive and easy-to-use, including business logic." [21, p. 26]. Design that attends to these issues might better predict and avoid errors, enable more flexibility to accommodate users with different skills levels, better enable consistent design and software standards, and better accommodate real-world needs and workflows.

3.3 Software Review Findings

The web findings were analyzed by using Nielsen's heuristics for usability. The ten heuristics are a means by which to assess the accessibility and usability of products. They include the following: system visibility, match between the system and the real world, user control and freedom, consistency and standards, error prevention, recognition rather than recall, flexibility and efficiency of use, aesthetics and minimalist design, help diagnosing and recovering from errors, and help and documentation. Ratings were compared with narrative descriptions of the users' experiences of the products, then they were divided into categories based on the ten heuristics (Table 3).

Table 3. Rating scores per heuristic.

	Overall	Ease of use	Customer service
Visibility of system status	4.5	4.25	4.75
Match between system and the real world	3.8	3.5	3.8
User control and freedom	3.6	3.4	3.7
Consistency and standards	2	1.7	1.7
Error prevention	3.6	3.4	3.6
Recognition rather than recall	3.5	3.3	4
Flexibility and efficiency of use	4.0	3.9	4.1
Aesthetic and minimalist design	2.9	2.5	2.6
Help users recognize, diagnose, and recover from errors	–	–	–
Help and documentation	3.3	3	3.1
Overall Review Scores	4.5	4.3	4.6
Negative Review Scores	4.1	3.9	4.2

Of 150 reviews collected, there were 65 negative reviews with several suggestions for improvements or complaints about limitations. Even though 43.3% of the reviews contained suggestions and complaints, the overall rating average was 4.5 out of 5. Surprisingly, the sub-average of rating for suggestions and complaints was only slightly lower: 4.1 out of 5. Comments about poor help support and documentation, difficulty escaping and avoiding error messages, software inconsistencies, difficulty finding instructions and definitions, repetitive errors or pain points, and poor interface design earned the lowest overall scores, as well as lowest scores for ease of use and customer service. On the other hand, despite the fact that there were more comments about problems with software flexibility and customization than any other feedback, these overall scores, as well as scores for ease of use and customer service, were high.

Most of the suggestions and complaints were about the inability for tailoring software to facilitate frequent actions and meet needs (n = 39). For example, one comment mentioned software limitations when it comes to accommodating their policies: "Limited options for complex…policies & only has bi-weekly accruals." On the other hand, comments also indicated some disadvantages to customization, such as favoring some users over others: "There are times when suggested changes cannot be implemented without affecting all users." Several suggestions and complaints also pertained to poor design leading to errors (n = 20). For example, one comment reported great difficulty with simply setting up the software: "The setup that is done for my application does not work for me."

The language used within incident reporting software was often counterintuitive, and the cost was often exorbitant. These suggestions and complaints spotlighted the mismatch between the system and practical use (n = 19). For example, software often did not populate the correct real-world address of events: "Sometimes the addresses that the residents give us will not come up on the prepopulated drop-down menu. Most of the time the problem is the address may be a store within a building that has a lot and block that are for the entire building and not just that store." There were many suggestions and complaints about the limitations and poor quality of help and tech support (n = 17). For example, several comments criticized poor tech support, customer service, and help documentation: "I contacted Customer Service and it took them 2 weeks to get back to me for a simple database question." There were fewer comments about problems with visibility (n = 4), consistency (n = 3), recognition (n = 3) and aesthetics (n = 5). For example, the software often was said to have "a bit of an older look and feel." Overall, a few comments suggested that the interface design was old and unfriendly and that more—and more intuitive—dialogue boxes would make the software more usable (Table 4).

Table 4. Integrating the findings.

	Focus group	Prior literature	Reviews	Total
Visibility of system status	–	–	4	4
Match between system and the real world	–	4	19	23
User control and freedom	27	1	11	39
Consistency and standards	–	1	3	4
Error prevention	37	6	20	63
Recognition rather than recall	–	–	3	3
Flexibility and efficiency of use	24	3	39	66
Aesthetic and minimalist design	19	5	5	28
Help users recognize, diagnose, and recover from errors	6	2	–	9
Help and documentation	33	1	17	51

4 Conclusion

Incident reporting software for fire services and first responders needs improvement overall. Three different sources—user feedback from interviews, prior literature, and ratings online—revealed several problems with the responsiveness, expensiveness, and accessibility of the software. The most common problems mentioned across all three sources were problems with error-prone conditions built into the design that trigger problems (n = 63), a lack of or too much of flexibility and efficiency of use (n = 66), and poor-quality help and documentation (n = 51). Problems with users being unable to control and escape program functions (n = 39), interface design that was either unhelpful or outdated (n = 28), and mismatches between the system and the real world needs and limitations (n = 23) were also dominant.

These findings suggest that contemporary best practices in user-centered design might very easily resolve the problems with incident reporting software. Several times (n = 37) in the user focus groups, participants compared their incident reporting software to other, better software that they use on the job. This fact only goes to show that there are, indeed, contemporary solutions to database programming, data transfer, and cross-platform usability that would benefit and update the software in this sector. However, the limitations of service sector budgets might be a barrier to applying these solutions to incident report writing software products.

Findings from this review also suggest that poor visibility of system status, lack of match between system and real-world use, and opaque help and documentation are common barriers. Interface design complicated cross-platform use from computer to mobile devices, and systems often did not provide adequate user feedback. NFIRS software programs also do not anticipate the cultural idiosyncrasies endemic to fire services (such as apprenticeship learning, log lag, house rules, and level of education) that, if addressed, could help software users better recognize, diagnose, and recover from decision-making errors. Firefighters request more functionality for reporting medical calls, finding pertinent codes, differentiating between nondescript codes, data

management, and interface responsiveness. Artificial intelligence and responsive design could be used to develop solutions that help mitigate these issues.

These findings also shed some light on potential confounders in incident report writing. The software could do a better job providing automation or help documentation to help clarify unclear reporting standards. Software help documentation and services could help fill in the gap in report training and quality control processes. Improving help documentation could also help users better navigate the complexity of the NFIRS coding schema. An error-sensitive design could help minimize time on task, as could better mobile options that would allow reporting to happen on site.

Recommendations for improving the quality of software programs for incident reporting in fire services include, among others, improving customization features, providing templates and content guides, and improving the glossaries of common acronyms. Help systems should also enlist multimodal content to accommodate the diverse backgrounds and levels of education that comprise fire services.

Acknowledgments. The project was funded by State Farm Companies Foundation.

References

1. Hopia, H., Latvala, E., Liimatainen, L.: Reviewing the methodology of an integrative review. Scand. J. Caring Sci. **30**(4), 662–669 (2016)
2. Whittemore, R., Knafl, K.: The integrative review: updated methodology. J. Adv. Nurs. **52**(5), 546–553 (2005)
3. Cooper, H.M.: Scientific guidelines for conducting integrative research reviews. Rev. Educ. Res. **52**(2), 291–302 (1982)
4. Torraco, R.J.: Writing integrative literature reviews: using the past and present to explore the future. Hum. Resour. Dev. Rev. **15**(4), 404–428 (2016)
5. Doolen, J.: Meta-analysis, systematic, and integrative reviews: an overview. Clin. Simul. Nurs. **13**(1), 28–30 (2017)
6. Smith, M.C., Stullenbarger, E.: A prototype for integrative review and meta-analysis of nursing research. J. Adv. Nurs. **16**(11), 1272–1283 (1991)
7. U.S. Fire Administration. National Fire Incident Reporting System active vendors. https://www.usfa.fema.gov/data/nfirs/vendors/active_vendors.html. Accessed 1 Feb 2019
8. U.S. Fire Administration. National Fire Incident Reporting System registered vendors. https://www.usfa.fema.gov/data/nfirs/vendors/registered_vendors.html. Accessed 1 Feb 2019
9. Kinsey, K., Aherns, M.: NFIRS incident types: Why aren't they telling a clearer story? National Fire Protection Association, Quincy (2016)
10. Stefancic, J.: NFIRS Data Entry Analysis, Homeland Security Digital Library (2011). https://www.hsdl.org/?view&did=698931. Accessed 1 Feb 2019
11. Federal Emergency Management Agency (FEMA): Conquering the unknowns: research and recommendations on the chronic problem of undetermined and missing data in the causal factors sections of the National Fire Incident Reporting System, The Foundation, Washington, DC (2013)
12. Smith, M.: Determining NFIRS reporting accuracy by Alachua County Fire Rescue company officers. National Fire Academy, Emmitsburg (2007)
13. Stefancic, J.: Fire Inspection Program Analysis. National Fire Academy, Emmitsburg (2010)

14. Nielsen Norman Group: 10 Usability Heuristics for User Interface Design (1994). https://www.nngroup.com/articles/ten-usability-heuristics/. Accessed 1 Feb 2019
15. Krueger, J.: Garbage in? Evaluating the consistency of the data input into the MTFPD NFIRS compliant database. National Fire Academy, Emmitsburg (2010)
16. Coleman, K.J.: Are low cost accountability, communications, and management systems for emergency first responders using 3G and 4G cellular technologies feasible? Rochester Institute of Technology, Rochester (2010)
17. Peacock, R., Forney, G.: Multi-phase study on firefighter safety and the deployment of Resources (2008)
18. Romano, M., Onorati, T., Aedo, I., Diaz, P.: Designing mobile applications for emergency response: citizens acting as human sensors. Sensors **16**(3), 406 (2016)
19. del Olmo Pueblas, S.: Real-time visualization of multi-source, heterogeneous data in emergency. Association for Computer Linguistics, Stroudsburg (2015)
20. Chronaki, C.E., Kontoyiannis, V., Charalambous, E., Vrouchos, G., Mamantopoulos, A., Vourvahakis, D.: Satellite-enabled eHealth applications in disaster management-experience from a readiness exercise. In: Computers in Cardiology, pp. 1005–1008 (2008)
21. Averill, J.D., Moore-Merrell, L., Notarianni, K., Santos, R., Wissoker, D.: Proceedings of the National Fire Services Data Summit (No. Technical Note (NIST TN)-1698) (2011)