International Journal of Asia Digital Art & Design

Received March 1, 2016; Accepted April 2, 2016

A pattern language: designing a hazard information map interface for community-based users

Haimes, Paul
Tokyo Metropolitan University
haimes@tmu.ac.jp

Medley, Stuart
Edith Cowan University
s.medley@ecu.edu.au

Baba, Tetsuaki
Tokyo Metropolitan University
baba@tmu.ac.jp

Abstract

This paper explains guidelines that arose from a collaborative research project that aimed to make map-based bushfire information more accessible to people in remote and regional Australian communities. The MyFireWatch web application was a practical outcome from this research and was the result of several iterations of user and service-provider engagement. This application delivers a web-based interface that works on desktop and mobile devices which displays bushfire locations around Australia in near real-time. As a way of generalising from the work undertaken that resulted in the MyFireWatch application, guidelines were created to inform others working in similar domains. These guidelines are presented here in the form of a pattern language and are intended to inform the design of similar systems. Pattern language has previously been used in architecture, software engineering and interaction design as a means of exchanging knowledge in a way that provides specific solutions to recurring problems, yet these solutions are generalised so that they can be applied in different scenarios. The pattern language described here is intended to encourage others who may be working with map-based hazard information to consider various aspects of the interface and its functionality. It is hoped that in doing so, communities in Australia and beyond will find such systems more accessible, intuitive and easy to use.

Keywords: human factors, user interface guidelines, GIS map applications

1 Background

Landgate – the primary source of geographical data and land information in Western Australia – produces several fire monitoring services derived from multiple sources including near real-time satellite imagery, aerial photography and lightning strike detection. These services – referred to as FireWatch – were originally built for the use of emergency services professionals. In usability terms, FireWatch was stymied by its history as a service for technical users, in that its emphasis on providing a high level of technical data resulted in a lack of consistency and ease of use in the interface. The major aim of this research was to redesign a public access version (Figure 1) of FireWatch for the use of non-technical users in the wider community, using a trial regional community to fine-tune the service. The redesign aimed to present a more usable and intuitive interface for these non-technical users. The focus of the redesigned interface is on wider community use. The intention is to inform communities of actual and potential fire dangers in their community and assist them in making decisions of how to prepare and respond to these dangers. After several design iterations, which included direct input from users in rural communities, the redesigned interface was launched as an officially supported publicly-accessible web application known as MyFireWatch [1] in 2014. This research was undertaken as constructive design research, which "refers to design research in which construction – be it product, system, space, or media – takes centre place and becomes the key means in constructing knowledge. Typically, this 'thing' in the middle is a prototype" [2]. Part of undertaking constructive design research involves generalising from the specific in order to generate frameworks and guidelines for others who may be working in a similar realm [2].

Hence, a framework is presented here – in the form of a pattern language – that generalises from the MyFireWatch design process and results generated by this research so that those aiming to present hazard information to community-based users – including the development team of

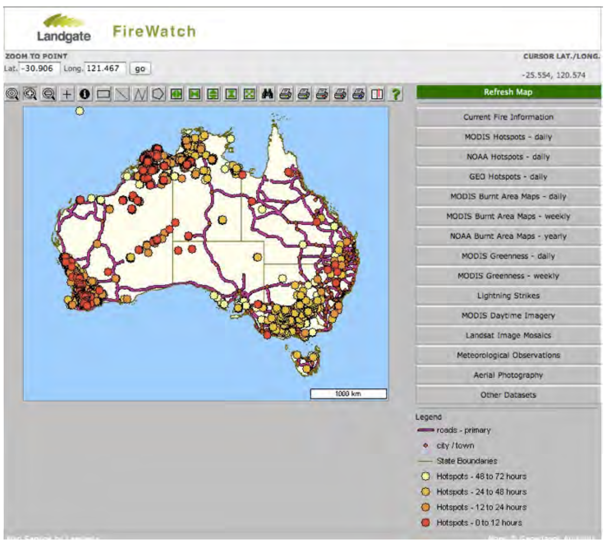


Figure 1: The original publicly-accessible version of the FireWatch application.

the industry partner Landgate – have a starting point for undertaking their own design work. These design patterns focus on improving both usability and utility of an interface and were formed as a result of multiple rounds of user input through various methods including semi-structured interviews, observations, testing of the interface (The final version is shown in Figure 2) and a usability questionnaire.

2 Pattern Language

Created by Alexander et al as a language for describing solutions to problems identified in architecture [3], pattern language found its way into software engineering and HCI in the 1990s [4, 5]. Interaction design has continued this practice [6]. Design patterns improve on style guides and standards as a way to express interaction design experience [5]. Patterns provide interaction designers with a means to a concrete example and a generalised solution while also offering a context in which to apply the solution [5]. A pattern language also has a hierarchy, which "leads the designer from patterns addressing large-scale design issues, to patterns about small design details" [5]. In interaction design research, design patterns serve the purpose of formalising design knowledge and documenting best practices [6]. Design patterns serve to reduce design time and effort on new projects, can improve the quality of design solutions, facilitate communication between designers and programmers and educate designers [6]. Borchers noted that Alexander's intention with pattern language was "to allow not architects, but the inhabitants (that is, the users) themselves to design their environments. This is strikingly similar to the ideas of user-centered and participatory design" [5]. For this reason, Borchers noted that design patterns acted as a universal language amongst members of interdisciplinary design teams [5].

2.1 Working with users of the MyFireWatch system

The pattern language explained here arose as a way of generalising outcomes from the final MyFireWatch design iteration, which was the result of two rounds of user testing, observations and interviews, plus a usability questionnaire where online users provided feedback on the final design iteration of the MyFireWatch system (Figure 1). In the two rounds of user input, users (n=17) were first asked to rate - using a card system [7] - the following features provided by the MyFireWatch system in terms of usefulness (Very useful, somewhat useful, somewhat non-useful, very non-useful):

- Aerial view of the terrain (satellite view)
- Previously burnt areas
- Current fire hotspots
- Greenness of vegetation
- Lightning strikes
- Location search
- Weather data [7].

Users were able to elaborate on their ratings and were also asked whether there were any additional features not provided that they thought would be useful. Only features that had an average rating of "very useful" or "somewhat useful" were included in the interface. Users then spent several minutes using and evaluating the interface before being asked a series of questions in a semi-structured interview regarding usability of the interface and whether the functionality provided met their needs [7, 8]. The input from users in these two rounds of user input directly informed the design of the MyFireWatch interface (Figure 2), including the choice of features included in the interface, the placement of these features and how these were presented [7, 8].

The final version of the MyFireWatch interface was then made publicly available [1]. Coinciding with this new publicly accessible version of MyFireWatch was the online launch of a usability questionnaire, based on the System Usability Scale (SUS) [9]. An interface that scores more than 68 out of 100 on the SUS is considered a usable interface [10]. The averaged results amongst responses (n=34) to the online usability questionnaire was 74 out of 100, meaning that it provided the majority of users with a usable interface which met their needs. 74% of respondents found the application easy to use. 83% of respondents said that it was quick to learn how to use the application. 62% of respondents said they would use the application frequently. Less than 1% of respondents found the application unnecessarily complex and that there was too much inconsistency in the design. These positive results verified that the design process undertaken in two rounds (user testing, including a card rating system, observations and a semi-structured interview [7, 8]) with users resulted in a usable interface that met the needs of the majority of community-based users.

2.2 Structure and use of the patterns

The pattern language described here consists of 17 individual design patterns to guide designers, researchers, application developers and GIS (geographical information system) experts working with map-based hazard information. As a way of generalising from the results generated through the two rounds of user input and the online questionnaire [7, 8], these patterns were created as a way to guide others working in a similar area. The patterns were also created to assist Landgate in the event that they design or redesign other similar map-based products that present hazard information to community-based users.

The patterns are grouped into three categories, based on the categories of the requirements used to guide the design iterations: functional requirements, data requirements and other requirements [6]. Definitions of each of these categories are included in the pattern language below. Grouping requirements into these three categories proved to be effective when establishing requirements at the beginning of each iteration of the redesign process that resulted in the MyFireWatch application [8]. They are therefore used here for categorising the design patterns. Following the requirements described in these design patterns is likely to provide a strong foundation for a map-based hazard information application that is both useful and usable for community-based users.

The patterns follow the same structure Borchers [5] described within the context of HCI and interaction design, with each pattern comprising a name, problem, context, solution, example, references (to other relevant patterns within the pattern language), and – where appropriate – a diagram of an example solution. The solutions provide a general way of addressing the identified problem, while examples provide a specific example of how the problem has been previously addressed. Examples also include references to user feedback acquired in the user testing and questionnaire described in the previous section. The example diagrams are taken from the final version of the MyFireWatch interface. These components

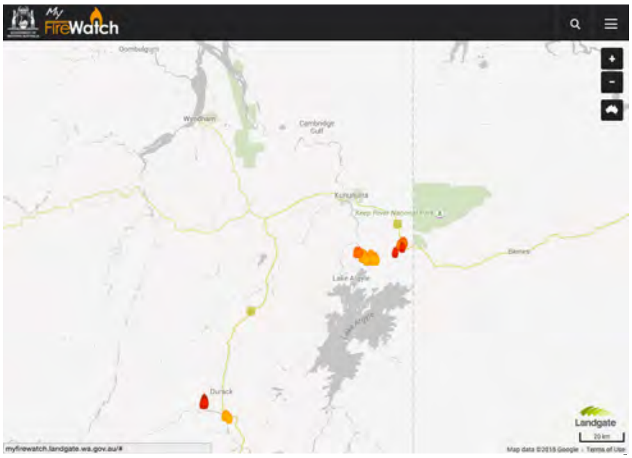


Figure 2: The mobile interface of the MyFireWatch application [1], which was launched in 2014.

are considered essential to design patterns [5]. For each design pattern presented in the pattern language described here, evidence is provided from the literature that guided the design process, as well as results that arose from the user input explained in the previous section. For ease of use and quick reference, the pattern language explained here only has two levels in the hierarchy (See Tables 1, 2 and 3). Level one patterns describe generalised, higher level guidelines while level two patterns refer to more specific features of the interface.

Table 1: Hierarchy of *functional* patterns

Pattern name	Hierarchy level
Simplicity	1
Consistency	1
All Devices	1
Geo-locate	2
Search	2
Zoom	2
Map	1

Table 2: Hierarchy of *data* patterns

Pattern name	Hierarchy level
Map navigation	1
Default information	2
Alerts	2
Satellite view	2
Other layers	2
Layer options	2

Table 3: Hierarchy of *other* patterns

Pattern name	Hierarchy level
Information source	2
Simple language	1
Minimal download time	1
Natural mappings	1

These patterns can serve as a starting point for establishing functional, data and other requirements prior to commencing an interface design for a hazard information system. However, as the individual patterns contain a significant level of detail and a specific example (and reference to a diagram where appropriate), they provide a solid foundation of how each requirement can be addressed, serving as a blueprint for how each requirement can be designed.

2.3 Functional patterns

Functional requirements are those which address "operations or actions that need to be performed on the objects of the system and which are typically translated into interface controls" [6]. These requirements include generalised principles for how to address the interface design, as well as the essential interface components.

Pattern name: SIMPLICITY

Context: A simple interface will be more usable. Simplicity is important, particularly when presenting hazard information to users.

Problem: Information related to hazards needs to reach the user quickly but also needs to be easy to understand [11].

Solution: Only provide features that are crucial to the application. Identify the core features. Consider whether or not to include sub-features. Any complex aspects need to be managed by a designer to make them easily understandable [12]. However, "when in doubt, remove" [13].

Examples: The original expert-user version of FireWatch provided several datasets for current fires. This included map layers labelled "Current Fire Information", "MODIS Hotspots – daily", "NOAA Hotspots – daily", "GEO Hotspots – daily" and "NPP Hotspots – daily". MyFireWatch contains only one set of current fire hotspots, labelled "Current fires". This single set of hotspots met the needs of the community-based users who undertook user testing [7, 8].

References: The MAP feature should initially display DEFAULT INFORMATION only. OTHER LAYERS should be available, but these should only use minimal LAYER OPTIONS.

Pattern name: CONSISTENCY

Context: Elements that perform a similar function should be consistent in their actions and aesthetics.

Problem: Interface objects should be consistent with their behaviour. Objects with different behaviour should appear differently [14]. Users also prefer interfaces that they are familiar with [15].

Solution: Elements that perform the same type of function – such as the main site navigation and the map navigation – should have the same appearance and perform in the same way. They should use the same font, same font size and colour. If icons are used, they should be around the same size, and where appropriate, use the same or similar colours. Navigation should behave the same when clicked or touched. Web users in general seek familiarity and consistency in the interfaces they use [16], so follow conventions found in other map applications that users will be familiar with, such as Google Maps.

Examples: The MyFireWatch application's main navigation and map navigation's text and link behaviour are the same. The map navigation had different icons for each map layer, but these icons were approximately the same size, and similar layers (such as the four hotspot layers) were the same size. 16 of the 17 participants who provided input through user testing and interviews found the interface easy to understand.

References: Only use SIMPLE LANGUAGE in the interface. DEFAULT INFORMATION, OTHER LAYERS and LAYER OPTIONS should all be consistent in their appearance and function. Consider NATURAL MAPPINGS in the context of the visual aspects of the interface.

Diagrams: See Figure 4 and Figure 6.

Pattern name: ALL DEVICES
Context: The interface should work on mobile and tablet devices as well as desktop.

Problem: A modern interface needs to cater for several screen sizes, from a large monitor to the portrait orientation of a small smart phone. It should work on all common web browsers.

Solution: Use responsive design (Marcotte, 2011) to cater for all screen resolutions. All content should be flexible in its width. Wherever possible, use percentages rather than pixel values for an element's width. For example in CSS, set the content area's width to 100%. Text should be readable on all devices by using font sizes around 16 pixels and maintaining high contrast between the background and font colour.

Examples: In the MyFireWatch interface, the top area's width stretches across the entire width of the screen. This area adjusts according to the size of the screen being used, ensuring that it is easily accessible regardless of which device a user accesses the application with.

References: The MAP component will be the most prominent feature of the interface.

Pattern name: GEO-LOCATE
Context: The map should automatically detect the user's location.

Problem: The user needs to easily locate fires in or near their location.

Solution: Use geo-location functionality [17]. This feature is included in most modern browsers and smart phone operating systems.

Examples: MyFireWatch automatically detects the user's location using HTML 5's automatic geo-locate. This feature makes it easier for the user to orientate themselves [7].

References: SEARCH and ZOOM also allow the user to orientate themselves, but GEO-LOCATE automates this process.

Pattern name: SEARCH
Context: Users need to search for a location by address, a town name or postcode.

Problem: Users need to contextualise the information provided to a location of their interest. To do so, they want to search by information such as postcode, address or town name.

Solution: Allow users to enter information in an easy to understand format such as postcode, address or town name [7]. The search bar should be easy for the user to locate and should be displayed prominently above the map.

Examples: MyFireWatch allows users to enter location information in a variety of ways, including an address, a town name, postcode or latitude and longitude. The search bar is prominently displayed above the map feature. As several users overlooked the search feature in early iterations of the design, it was made wider and more prominent in the final design iteration.

References: GEO-LOCATE can automatically detect a user's location. ZOOM can allow a user to display the map at a resolution that is useful to their personal circumstances.

Diagram: Figure 3.

Figure 3: The search functionality provided by the MyFireWatch application. This feature was placed directly above the map and took up more than two thirds of the screen width. As some users overlooked this feature in early iterations of the design, it was made more prominent in the final design iteration.

Pattern name: ZOOM
Context: Users require controls to zoom in and out on the map.

Problem: Users need to scale the map feature to a resolution that is meaningful to their personal circumstances.

Solution: Add zoom controls in the top right corner of the map. These controls should be big enough to be easily accessed on a mobile or tablet device. A plus and minus sign have become de-facto standard ways of visualising this feature (e.g., [16, 17]).

Examples: MyFireWatch includes zoom controls in the top right of the map. These controls are displayed prominently and were big enough to be easily accessed by users who tested the application on mobile and tablet devices.

References: GEO-LOCATE and SEARCH can also allow the user to contextualise the information to a location of interest.
Diagram: The zoom functionality is visible in the top right corner of Figure 2.

Pattern name: MAP
Context: The map is the main focus of the application.
Problem: Spatial information related to the hazard should be displayed clearly and simply.

Solution: The map should be the largest component of the interface, using proportion to draw attention to it in relation to other elements [20]. The width of the map should take up most of the browser's width. The map itself should show town names, names of national parks and roads [7].

Examples: In the MyFireWatch interface, the map's width stretches to the right-hand side of the screen. On the desktop version, there is a left margin of 236 pixels to make room for the layer navigation. On the mobile version the map stretches to the left-hand side of the screen, meaning that the map component has a width of 100% of the screen.

References: MAP NAVIGATION is required to toggle the map layers – including DEFAULT INFORMATION and OPTIONAL LAYERS – on and off.

Diagram: The default map view is visible in Figure 2.

Pattern name: MAP NAVIGATION
Context: Controls are required to allow the user to toggle map layers on and off. The default hazard information is switched on by default.

Problem: Users require access to the information provided, and the ability to toggle the map layers on and off. Users also need to easily understand what the layers mean.

Solution: The navigation should allow users to toggle layers on and off. Where appropriate, the navigation icons can act as a legend for the map. Use both text and icons – multimodal communication – to portray the meaning of the layers. Simple terms should be used when labelling the layers. On the desktop version of the interface, the map navigation should be situated to the left of the map. This is the case with well-known map applications, such as Google Maps [18] and Bing Maps [19], as well as MyFireWatch (desktop version).

Examples: The MyFireWatch application provides navigation controls for every layer. The navigation is displayed to the left of the map feature on the desktop interface and is accessible via the dropdown menu on the mobile and tablet interface. Both text and icons are used to convey the meaning of the layers to the user. Only simple terms are used, such as "Current fires" and "Burnt areas". These terms were easily understood by users of MyFireWatch: no user who undertook user testing had problems understanding the terms used. In the case of the current fires, burnt areas and lightning activity, the navigation icons also act as a legend. For example, the fire hotspots navigation also acts as a legend – telling the users which icons refer to which timeframe.

References: The map navigation is used to control which map layers are displayed on the MAP. The navigation should only use SIMPLE LANGUAGE. Adding GEO-LOCATE makes it easy for the user to orientate themselves. Consider NATURAL MAPPINGS when creating icons for the map navigation.

2.4 Data patterns
Data requirements are those which discuss "objects and information that must be represented in the system" [6]. Here, the guidelines describe information provided by the interface that is related to hazards.

Pattern name: DEFAULT INFORMATION
Context: A hazard map should show only the hazard information by default.

Problem: Information related to hazards needs to be provided to the public. The information needs to be "timely and understandable to those at risk" [21].

Solution: The hazard information should be the only layer that appears on the map by default. This will make the information easy to understand for users at risk. Use of colour should be appropriate: for example, if there are common ways of representing a particular kind of information, that standard should be followed.

Examples: The MyFireWatch application only shows current bushfire information by default, on top of the map layer. Other information is accessed through the map navigation to the left of the map or by the dropdown menu on mobile and tablet devices. Layers that are currently shown on the map display a tick and a grey background. The colours of the icons use the same spectrum of colours used in the previous expert-user version of FireWatch (Figure 1) and Sentinel [22].

References: The MAP should be the most prominent feature of the interface. ALERTS and OTHER LAYERS can provide additional information to users. GEO-LOCATE, SEARCH and ZOOM can assist users in displaying the information at a resolution that is meaningful. Consider NATURAL MAPPINGS when creating icons for the default layers.

Diagram: Figure 4.

Figure 4: The default fire hotspot information provided by the MyFireWatch application. These are the only layers displayed when the interface initially loads. Only the essential hazard information should be displayed initially. Layers that are currently shown on the map display a tick and a grey background.

Pattern name: ALERTS

Context: Alert information should be added where available in addition to the default hazard information.

Problem: Users seeking more information about hazards will look to official alerts for further information.

Solution: Where possible, provide information from – or links to – official feeds of emergency services organisations. If the alerts are geo-tagged, they can be added to the map. Otherwise they will need to be displayed separately to the map feature [7].

Examples: The MyFireWatch application provides an Alerts page accessed by the main menu. The alerts are also linked from the popups accessed by clicking or touching the fire hotspots. Although few MyFireWatch users requested that alerts be provided, this feature was added due to a request from the service provider, Landgate. The information is provided by state and territory-based emergency services organisations with external links to those organisations. Where available, links to an organisation's Twitter page were also provided.

References: By default, the MAP should only provide DEFAULT INFORMATION. This default information can be supplemented by alerts and OTHER LAYERS.

Pattern name: SATELLITE VIEW

Context: Provide real imagery of the land by providing users with a satellite or aerial view of the terrain.
Problem: Users require real imagery of the land to orientate themselves to key features in the landscape at closer zoom levels [8].

Solution: Provide an aerial or satellite view that provides real imagery of the terrain.

Examples: In the MyFireWatch application, a satellite view is available under the "MAP OPTIONS" heading in the map navigation. MyFireWatch uses the satellite view from Google Maps. Participants who took part in user testing were more easily able to locate key features in the landscape when this layer was visible at closer zoom levels (i.e., larger scales) [8].

References: The satellite view should be one of a few OTHER LAYERS provided to users.

Diagram: Figure 5.

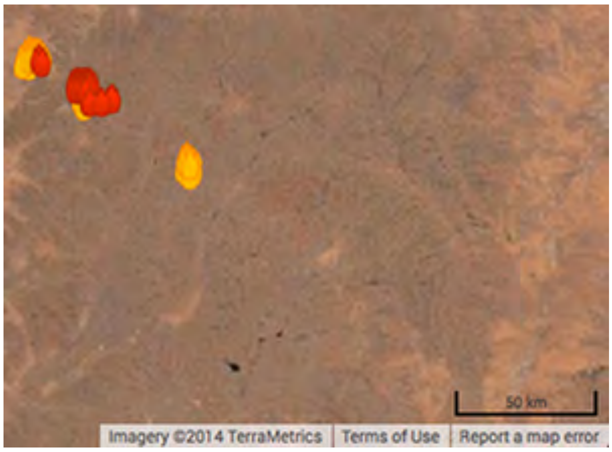


Figure 5: The satellite view provided by the MyFireWatch application. The application uses satellite imagery from Google Maps. Users require this realistic view of the terrain in order to orientate themselves to key features in the landscape.

Pattern name: OTHER LAYERS

Context: Provide some additional layers to supplement the default hazard information. These layers should not be visible by default.

Problem: Additional layers should be provided to users to supplement the default hazard information. This additional information can assist users in decision-making related to the hazards in their vicinity.

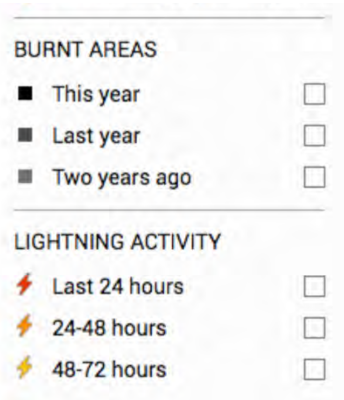
Solution: Provide additional information that is related to the hazard. This should include a satellite or aerial view of the terrain. Care should be taken to only provide information that is useful for non-expert users [7]. Relevant supplementary information may include historical data and meteorological information.

Examples: MyFireWatch provided users with five other types of information in addition to the fire hotspots: a satellite view, weather, greenness of vegetation, burnt areas and lightning activity. The inclusion and positioning of these layers was determined by feedback acquired from the card system used during user testing [7, 8].

References: Only provide up to three LAYER OPTIONS to users. Amongst these other layers should be a SATELLITE VIEW. Remember to consider NATURAL MAPPINGS when using icons.

Diagram: Figure 6.

Figure 6: The other layers provided by the MyFireWatch application. Care should be taken to ensure that these layers only provide information that is considered essential to a community-based, nontechnical audience.



Pattern name: LAYER OPTIONS
Context: Additional information on the map interface should only provide up to three options.
Problem: Information in addition to the default hazard information needs to be provided to users, to assist in decision-making. However, care needs to be taken to not overburden the user with too much information.

Solution: Provide up to three options for each type of additional information. This amount of information should meet the needs for the majority of community-based users [7].

Examples: MyFireWatch provides three map options to users: a satellite view, weather and greenness of vegetation. There are three years of burnt areas and three days of lightning activity. These layers met the needs of the majority of community-based users who undertook user testing [7, 8].

References: OTHER LAYERS should be provided, in addition to the DEFAULT INFORMATION.

Diagrams: See the diagram provided in the OTHER LAYERS pattern (Figure 6).

2.5 Other patterns
According to Cooper et al, other requirements can include things such as business, brand or technical requirements [6]. Here, they describe technical requirements, how to present technical information to non-technical users and also address issues such as credibility and use of icons.

Pattern name: INFORMATION SOURCE
Context: Information about the source of the hazard data should be available to users.
Problem: Users need to know the source of the hazard information being provided.

Solution: Information about where the data comes from should be provided. The information provided should be timely and relevant to the user. The navigation should also be easy to use. These aspects will add to the credibility and trustworthiness of the application [23]. If necessary, provide disclaimers if the data comes form external providers. The information source should be a known organisation and the source of the information should be made known to the user [23].

Examples: In the MyFireWatch application, Landgate is clearly the organisation responsible, as their logo appears on every page. Knowing that Landgate – a known state authority in WA – owns the application adds to its credibility and authority [23]. Participants who undertook user testing did not doubt the source of information. There is also information about the source of the data – and its limitations – on the Terms and Conditions page, the About page and on the main landing page of FireWatch.

References: The source of the information should be described in SIMPLE LANGUAGE. MAP NAVIGATION should be easy to use.

Pattern name: SIMPLE LANGUAGE
Context: The terms used in the interface should be easy to understand for a non-technical audience.
Problem: The interface needs to be easily understood by a non-technical audience. Complicated terms, such as industry-specific information, can confuse a non-technical audience.

Solution: Avoid jargon – it is likely that community-based users will not be familiar with many of the terms used by professionals. Use only simple language. The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-based terminology [15].

Examples: In the redesign of the expert-user version of FireWatch (Figure 1), which resulted in the community-user focused MyFireWatch interface (Figure 2), all jargon was removed. This included terms such as references to satellites (e.g., NOAA, MODIS, etc.) as these terms were unlikely to be familiar to community-based users.

References: Features of the interface, such as SEARCH and MAP NAVIGATION should use simple language.

Pattern name: MINIMAL DOWNLOAD TIME
Context: The application needs to be quick to load [16].

Problem: Many users will be accessing the application by a mobile or tablet device, and therefore may be reliant on a slower connection.

Solution: Only provide information that is essential to community-based, non-technical users. By default, only provide the minimum hazard information required when the map interface loads. Since users on mobile and tablet devices may have limited internet access, restrict the number of layers that they can access initially. Note that the application's performance will be affected by the capacity of the server it is on, which may be beyond the control of those working on the application.

Examples: In MyFireWatch, the fire hotspots are the only map layers displayed by default. Additional layers only load when accessed via the map navigation. On the mobile and tablet interface, only one day of lightning activity and only one year of burnt area data are available.

References: The MAP feature should initially display the DEFAULT INFORMATION only. OTHER LAYERS should be restricted to those that are considered essential information for community-based users.

Pattern name: NATURAL MAPPINGS
Context: It should be obvious to users what effect the controls have on the system.

Problem: A design needs to be intuitive, ensuring that the spatial relationship between a system and its controls is as direct as it can be [24].

Solution: The icons used should represent their function obviously. This obvious representation reduces the cognitive load on the user [24]. The choice of icons should reflect the kind of information being provided and the type of actions being performed. A common example of this is using a "plus" symbol for zooming in on the map and a "minus" symbol for zooming out.

Examples: In the MyFireWatch application, the current hotspot map layers use flame icons to indicate the location of current fires. The choice of icons used for the map layer information (and the map navigation icons) relates directly to the type of information that the map layer provides and were understood by the majority of users who undertook user testing [7, 8]. The zoom controls for the map use plus and minus symbols for zooming in and out – a common way of providing this functionality (e.g., [18]).

References: When creating the MAP NAVIGATION, SEARCH and ZOOM features, consider natural mappings.

Diagrams: Consider the diagram under OTHER LAYERS (Figure 6). The icons used should relate directly to the kind of information provided by the map layers.

3. Further use and development of the pattern language

Currently, a project is being undertaken in Japan that aims to make disaster information more accessible to Japanese communities, that is, members of the public outside of emergency services personnel. The pattern language presented here served as a starting point for the prototyping of an iPhone application which shows active volcano warnings in near real-time [25]. In addition to serving as a set of requirements for the prototype, the pattern language was used to guide the development of each feature in this application such as searching, zooming and panning the map, and other navigational and data elements. This application serves as an example of how the pattern language presented here can be applied in a similar context. This iPhone application is currently undergoing user testing which will inform a future version of the pattern language described here. Early results (n=6) from this user testing suggest that the volcano warning iPhone application is not too complex, and easy to understand and use. Although user evaluation of this new application is ongoing, this early positive result suggests that referencing the pattern language described here will assist in creating an effective interface when building a similar system.
A copy of the pattern language has also been provided to the development team at Landgate as guidelines to refer to when developing future map-based services for community-based users.
Re-contextualising and evaluating these patterns through further user input will lead to further refinements of them.

4. Discussion and future work

The pattern language described here is considered a small step towards improving the communicative and informative aspects of internet-based hazard information available to communities vulnerable to such hazards. Others working in similar domains may use the pattern language as a starting point in their design process, which will then lead to further refinement of the patterns. As suggested by the user engagement described in the second section, community-based users looking to gain a more meaningful understanding of natural hazards may have a more satisfying user experience as a result.
Clearly the effectiveness of services such as MyFireWatch is highly dependent on the robustness and timeliness of the sensing technologies that detect hazards such as bushfires. As these technologies improve, so too will the usefulness of these services. The work described here means that as demand for and interest in these services grows, the interfaces will not be a barrier for users seeking such information.

5. Conclusion

This pattern language for hazard information systems deliberately avoids the specificity of software versions and industry jargon as these are constantly changing aspects of mapping software and various interaction contexts. The pattern language presented here rather revolves around user cognition, evolving out of results from two rounds of user input (through user testing, observations and interviews) and a user questionnaire. The desire to make an interface more intuitive for non-expert users overlaps completely with the need to provide a design process that does not have to change with every version of mapping software.

As noted by Borchers, communication amongst interdisciplinary design teams can be problematic for those who wish to communicate interaction design concepts and guidelines. Pattern language removes this communication barrier due to its universality. The example given here of a pattern language, and future examples that will refer to it and refine it, only strengthens Borchers' argument that pattern language ought to be a lingua franca for interaction design.

6. Acknowledgments

The first and second authors are grateful for funding for the MyFireWatch project from both the Australian Research Council (ARC) and Landgate, and wish to thank their ARC research colleagues at Edith Cowan University and Landgate. The Mapping Hazards in Japan research is funded by the Japan Society for the Promotion of Science.

References

[1] Landgate. MyFireWatch - bushfire map information Australia http://myfirewatch.landgate.wa.gov.au/, 2015.

[2] Koskinen, I., Zimmerman, J., Binder, T., Redstrom, J. and Wensveen, S. *Design Research Through Practice: From the Lab, Field, and Showroom*. Morgan Kaufmann Publishers Inc., 2012.

[3] Alexander, C., Ishikawa, S., and Silverstein, M. *A Pattern Language: Towns, Buildings, Construction*. OUP USA, 1977.

[4] Gamma, E., Helm, R., Johnson, R. and Vlissides, J. *Design Patterns: Elements of Reusable Object-Oriented Software*. Pearson Education, 1994.

[5] Borchers, J. A pattern approach to interaction design. *In Proc. of the 3rd conference on Designing interactive systems: processes, practices, methods, and techniques*, pages 369-378. ACM, 2000.

[6] Cooper, A., Reimann, R., & Cronin, D (2014). About Face 3: The Essentials of Interaction Design: Wiley.

[7] Haimes, P. *Designing for communities in bushfire-prone situations: Redesigning the FireWatch website interface*. (PhD), Edith Cowan University, Australia.

[8] Haimes, P., Brady, D., Clarkson, B., & Medley, S. (2013). *Engaging with communities as a design process: redesigning the FireWatch interface.* Paper presented at the Australian & New Zealand Disaster and Emergency Management Conference, Brisbane.

[9] Brooke, J. (1986). SUS – A quick and dirty usability scale. http://cui.unige.ch/isi/icle-wiki/_media/ipm:test-suschapt.pdf

[10] U.S. Department of Health & Human Services. (2012). Improving the User Experience. Retrieved from http://www.usability.gov/how-to-and-tools/methods

[11] Wu, P.F. Motivation for adopting emergency response technology in community settings. *In Proc. of the 22nd British HCI Group Annual Conference on People and Computers: Culture, Creativity, Interaction* - Volume 2, pages 259–260. British Computer Society, 2008.

[12] Norman, D. *The way I see it: Simplicity is not the answer. interactions*, 15(5):45–46, 2008.

[13] Maeda, J. *The laws of simplicity*. MIT Press, Cambridge, MA, 2006.

[14] Tognazzini, B. First principles of interaction design http://www.asktog.com/basics/firstPrinciples.html, 2014.

[15] Nielsen, J. Ten usability heuristics http://www.useit.com/papers/heuristic/heuristic_list.html, 2005.

[16] Nielsen, J. *Designing web usability*. New Riders, Indianapolis, 2000.

[17] Lawson, B. and Sharp, R. *Introducing HTML5*. Pearson Education, 2011.

[18] Google. About Google Maps. http://support.google.com/maps/bin/answer.py?hl=en&answer=7060, 2015.

[19] Microsoft Corporation. Bing Maps. https://www.bing.com/maps/, 2015.

[20] Evans, P. and Thomas, M. *Exploring the Elements of Design*. Thomson/Delmar Learning, Clifton Park, 2007.

[21] United Nations International Strategy for Disaster Reduction [UNISDR]. Hyogo framework for action 2005-2015: Building the resilience of nations and communities to disasters. http://www.unisdr.org/files/1037_hyogoframeworkforactionenglish.pdf, 2005.

[22] Geosciences Australia. Earth observation and satellite imagery - Sentinel. http://sentinel.ga.gov.au, 2015.

[23] Fogg, B. *Persuasive technology: using computers to change what we think and do*. Morgan Kaufmann Publishers, San Francisco, 2003.

[24] Norman, D. *The design of everyday things*. Basic Books, New York, 2002.

[25] Haimes, P. iTunes Preview: Katsukazan. https://itunes.apple.com/jp/app/katsukazan/id1084989877.