WeKnowIt

Emerging, Collective Intelligence for Personal, Organisational and Social Use

FP7-215453

D7.2

Emergency response and consumers' social group case study design and specification

Dissemination level	Public
Contractual date of delivery	Month 11, 28-02-2009
Actual date of delivery	Month 12, 03-02-2009
Workpackage	WP7 Case Studies
Task	T7.2.1 Development
Type	Report
Approval Status	Draft
Version	07
Number of pages	77
Filename	WP7_2009-02-16_v07_USFD_specification_template.doc

Abstract

This deliverable focuses on the Emergency Response (ER) case, where we first define the main functional and non-functional user requirements, with an in depth analysis of how organisational (ER response user) and personal intelligence (citizen user) requirements match or differ and how these user requirements can be met by an application that combines organisational and community intelligence.

A detailed description of the envisaged first prototype follows, with a description of the User Interface architecture followed by a description of

the main contribution that each workpackage (each intelligence layer) will provide and how this meets the user requirements.

We then present the design work that has been conducted by all the partners for the first prototype of the WeKnowIt application for ER use case, explaining the design principles that guided the work, followed by example mock-ups that refer to specific parts of the scenario defined in D7.1 or to the use case in general.

co-funded by the European Union

History

Version	Date	Reason	Revised by
01	24-10-2008	Product specification template from Lycos	Lyc
02	21-01-2009	Draft outline focused on ER use case	USFD
03	26-01-2009	Draft focused on ER use case with sections for partners contribution	USFD
04	06-02-2009	Integration of contributions from all the partners.	USFD
05	09-02-2009	New additions and document clean up	USFD
06	16-02-2009	Template fix	USFD
07	16-02-2009	Final editing	CERTH
08	28/02/2009	Integration of Partners contributions	USFD
09	02/03/2009	WP2 Contribution	CERTH
10	02/03/2009	Integration of Partners contributions	USFD
11	03/03/2009	Revised Conclusions	USFD

Author list

Organization	Name	Contact Information
USFD	Dr. V. Lanfranchi	vita@dcs.shef.ac.uk
USFD	Neil Ireson	nsi@dcs.shef.ac.uk
USFD	Gregoire Burel	G.Burel@dcs.shef.ac.uk
USFD	Elizabeth Cano	A.Cano@dcs.shef.ac.uk
UoKob	Dr. Ansgar Scherp	scherp@uni-koblenz.de
VOD	Costis Kontopoulos	Costis.Kontopoulos@vodafone.com
SCC	S. Slingsby	Sheila.Slingsby@sheffield.gov.uk
UoKob	Maciej Janik	janik@uni-koblenz.de
EM-KA	Andreas Sonnenbichler	andreas.sonnenbichler@em.uni-karlsruhe.de
EM-KA	Andreas Geyer-Schulz	Geyer-Schulz@iism.uni-karlsruhe.de
EM-KA	Michael Ovelgönne	michael.ovelgoenne@kit.edu
CERTH	Dr. Yiannis Kompatsiaris	ikom@iti.gr
CERTH	Dr. Phivos Mylonas	fmylonas@image.ntua.gr

Executive Summary

The advent of the Internet and of the user generated content sites as new media channels offer individuals the possibilities of instant sharing and broadcasting personal content (text, images, videos). People tend to communicate information and share content related to their individual experience and current situation, mostly for family, friends or known colleagues. However from the summation of this communication an overall understanding of events and trends may emerge.

This can be very useful in many applications including Emergency Response, where users share information about the events as they happen, or in shared social activities, where users share information about their activities and experiences. More specifically, Emergency Response is one of the case studies undertaken by the WeKnowIt Project to demonstrate the wide applicability of its methodologies and technologies and is the focus of the current version of this deliverable. The management of emergencies is very complex and challenging and Emergency Response Managers require speedy access to information to assist their response. Accurate and timely information is key to the effective response to emergencies and something identified amongst all emergency response partner organisations.

Currently emergency response applications are mainly based on personal and organisational intelligence and slowly expanding towards social intelligence. Usually these applications are targeted either at the Emergency Response personnel or at the citizens; the challenge of WeKnowIt is to provide a unified system that targets both citizens and emergency response personnel, providing different functionalities according to the user role. Additionally, by exploiting all the intelligence layers, Collective intelligence can help to match the user needs (of either emergency responders or citizens) by collecting and finding the needed knowledge at the right time and in the right modality.

In this deliverable, we first define the main functional and non-functional user requirements, with an in depth analysis of how organisational (ER response user) and personal intelligence (citizen user) requirements match or differ and how these user requirements can be met by an application that combines organisational and community intelligence.

A detailed description of the envisaged first prototype follows, with a description of the User Interface architecture followed by a description of the main contribution that each workpackage (each intelligence layer) will provide and how this meets the user requirements.

We will then present the design work that has been conducted by all the partners for the first prototype of the WeKnowIt application for ER use

case, explaining the design principles that guided the work, followed by example mock-ups that refer to specific parts of the scenario defined in D7.1 or to the use case in general.

The design process will continue following the iterative design methodology: sessions will be set up with users to walk through the mock-ups and provide feedback. The feedback will be incorporated in new mock-ups. It is envisaged that at least 2 iterations will happen before finalising the design of the first prototype.

Although this deliverable mainly concerns the work done for the Emergency Response use case, the methodologies and technologies adopted and the design process and results can easily be extended to other use cases, in particular to the Consumer Social Group case study, as they have been designed and developed in generic terms, thus making them easily customisable for specific domains.

Abbreviations and Acronyms

API	Application Programming Interface
CAP	Community Administration Platform
CDL	Community Design Language
CBRN	Chemical, Biological, Radiological and Nuclear
DUL	DOLCE+ Ultra Light
EPT	Emergency Planning team
ER	Emergency Response
FLO	Forward Liaison Officer
GUI	Graphical User Interface
JSF	Java Server Faces
JSP	Java Server Pages
IT	Information Technology
LVCSR	Large Vocabulary Conversational Speech Recognition
OOV	Out of Vocabulary
OS	Operating System
OWL	Ontology Web Language
PC	Personal Computer
PDA	Personal Digital Assistant
RANSAC	RANdom SAmple Consensus
RDF	Resource Definition Framework
ROI	Regions Of Interest
RVP	Rendezvous point
SCC	Sheffield City Council
SIFT	Scale Invariant Feature Transform
SMS	Short Message Service
SNA	Social Network Analysis
SOA	Service Oriented Architecture
SQL	Structured Query Language
SURF	Speeded Up Robust Features
UI	User Interface
UID	Unique Identifier
URI	Unique Resource Identifier
WP	Work Package
XRI	eXtensible Resource Identifier

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1. Purpose

The management of emergencies is very complex and challenging and Emergency Response Managers require speedy access to information to assist their response. Accurate and timely information is key to the effective response to emergencies and something identified amongst all emergency response partner organisations.

It has been demonstrated that increasingly the public are very willing to share information if they feel that they are helping the community and are keen to contribute to news events as they happen, for example terrorist attacks, severe weather disruption, train or air crashes, flooding, environmental issues. With the increasing use of camera phones, digital cameras, text messages etc the use of citizen involvement to assist in emergency response offers an exciting opportunity for emergency planners to utilise the citizens' contributions to improve the response to emergencies.

The use of new technologies to assist in their management of, and response to, emergencies would provide a means for emergency planners to access a wide range of information that is currently inaccessible. This information could be about a wide range of issues around an emergency such as affected roads, pedestrian routes, transport issues, evacuation centres, areas affected by an emergency such as flooding etc.

This information can be utilised by emergency planners to dramatically improve their understanding of the emergency and the emerging situation and will aid them in rapidly assimilating a large amount of information and assist in informing effective actions to prevent, contain or respond to a critical situation. It will enable emergency managers to quickly assess the best use of, and location for, scarce resources and therefore dramatically improve service delivery.

Of prime importance to emergency managers in utilising such a system will be its resilience and reliability. Any emergency response management system must be able to deal with and manage a large volume of information in a short period. The system should allow for information to be:

- Verified
- Prioritised
- Integrated with command and control structures
- Accessed at different levels

Information technology solutions to the provision of accurate, timely information generally fall into two categories: those aimed at Emergency

Response (ER) organisations and personnel, and those aimed at the citizens involved in the emergency. It is increasingly the case that individuals, given the instant publishing capabilities of Internet media, expect to be well informed during an emergency and ER organisations are required to respond to this demand. In addition, with the increased use of mobile devices and digital cameras, people have become accustomed to capture events and share the information and ER organisations are realising the potential offered by citizen involvement and content generation to provide useful information from the "ground", especially where the emergency is large-scale and widespread.

A critical factor in the successful development of an ER online content sharing system is to ensure that the requirements of both users (organisational and community) are met. While much work exists on the subject of ER technologies and applications [1] [2]; less research has been carried out on the User Requirements for applications involving citizens. In particular whilst there are a number of studies [3][4][5][6], which analyse user requirements with an emphasis on the ER organisational users, there is a gap when considering the generic public needs, and how they can match or differ from the ER organisational users ones.

In this deliverable we will introduce revised user requirements, with an in-depth analysis of the implied functionalities and services and a detailed description of the envisaged application.

1.2. Project Objectives

Currently emergency response applications are mainly based on personal and organisational intelligence and slowly expanding towards social intelligence. Usually these applications support only visualisation of the available information without any analysis and are targeted either at the Emergency Response personnel or at citizens; the challenge of WeKnowIt is to provide a unified system that targets both these target groups, providing different functionalities according to the user role. In addition, by exploiting all the intelligence layers, Collective intelligence can help to match the user needs (of either emergency responders or citizens) by collecting and finding the needed knowledge at the right time and in the right modality.

As WeKnowIt is aiming to exploit the power of Collective Intelligence to make a difference in real life situations, Figure 1 explains how the different Intelligence layers contribute to the creation of Collective Intelligence for the ER use case.

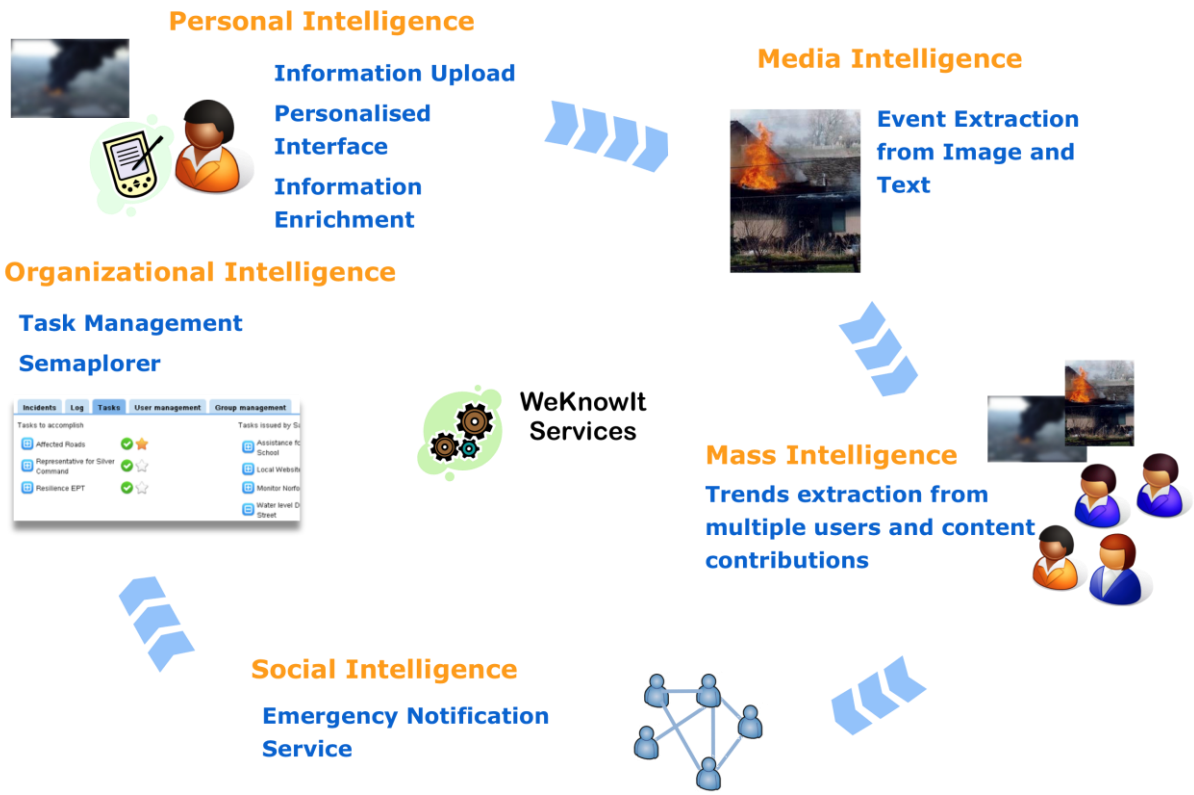


Figure 1 - Building Collective Intelligence

In order to ensure the communication and sharing between the content created and used by each workpackage in the project towards Collective Intelligence, a knowledge management methodology for Collective Intelligence has been devised in WP5, providing support for this methodology in the WeKnowIt project. The methodology shall allow for modelling and communicating collective knowledge gathered and processed from the different layers of intelligence, namely the personal intelligence, media intelligence, mass intelligence and social intelligence. The basic idea of our knowledge management methodology is to model the collective knowledge in terms of events. The rationale behind this is that humans think in terms of events. Thus, events provide a natural abstraction of happenings in the real world. Being very generic, events are independent from the different use cases considered in WeKnowIt. It allows structuring the collective knowledge gathered and processed by personal intelligence, media intelligence, mass intelligence and social intelligence. As shown in Figure 2, events can be used to enable sharing and interchange of the collective knowledge in the WeKnowIt project. A WeKnowIt Core System can be connected by events to different web services provided by the other layers of intelligence such as a service for image analysis, audio analysis, text classification, spatio-temporal clustering and others. By this, the event-based knowledge management methodology developed in WP5 serves as basis for enabling the interplay of the different layers of intelligence in the WeKnowIt project and enabling

the development of the two use cases of the project, namely Emergency Response and Consumer Social Group.

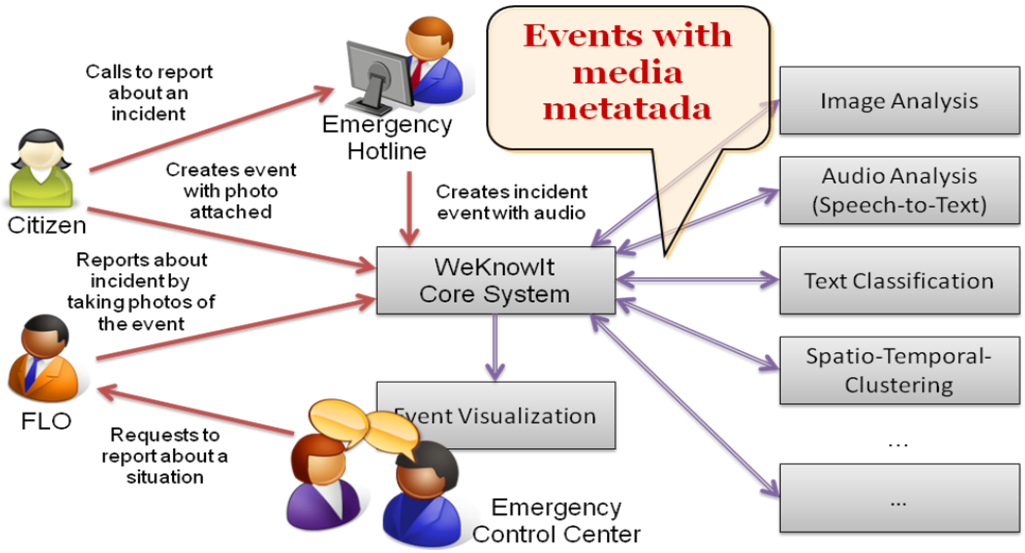


Figure 2 - Possible role of events in the WeKnowIt project

This methodology has then been adopted by WP1 to represent the possible objects and relations existing in the Personal Intelligence Toll and has been extended to the Application and Interface design in WP7, by mapping the use case scenarios defined in D7.1 with an event-based model and translating it into UI objects, as shown in Section 7.1.

2. User Requirements

In the following sections we will first highlight the target users considered for this application, and then introduce the user requirements derived from the user studies, analysed from different users perspectives.

2.1. Target Group

The target groups identified after the user studies are as follows:

- Citizens (intelligence from and to citizens will be conveyed through the personal intelligence layer)
- ER Professionals (intelligence to/from ER professionals will be conveyed through the organisational intelligence layer)

In order to better support the target group identified, the requirements outlined in [9] have been revised to understand the agreement and divergence between the user requirements of the two user-groups, and how these user requirements can be met by an application that combines organisational and community intelligence.

In the following sections we will detail the elicited user requirements, divided into functional and non-functional. In summary, an overlapping set of requirements was identified between the two user-groups, with both ER personnel and citizens citing the need for real-time ubiquitous access to relevant information during an emergency

2.2. Functional Requirements

Table 1 provides a summary of the agreement and divergence between the priorities (High, Medium, Low, or blank if not deemed relevant) given to the functional requirements by the two user-groups. As can be seen, the Organisational user generally has more strict requirements upon the system, as they generally have a greater vestige interest in the performance of the system, although this may not be the case for specific citizens.

Functional Requirement		Organisational Personnel	Community Citizen
MultiModal Interface	GUI, Touchscreen, Pen Input, Keypad (Mobile Phone/PDA)	H	H
	GUI, mouse-based interaction (PC)	H	H
	Speech Recognition	L	
Content Upload	Text	H	H
	Image	H	H

Functional Requirement		Organisational Personnel	Community Citizen
	Video	H	L
	Audio	L	
Information Enrichment	Geographic	H	H
	Temporal	H	L
	Importance/Priority	H	
	Semantic	M	
	Social	L	H
Search		H	H
Browse		H	H
Personalised Access	Multiple Users and Visualisations	H	M
	Information Control	M	
	Prioritisation/Filtering	H	L
	Alert	H	L
Recommendations		M	M
Checklist/ Task Management		H	
Feedback/Rating		M	L

Table 1 - Summary of User-group Functional Requirement Priorities

The key-requirements are now analysed in more details, explaining how they were derived and some example scenarios and how they can be achieved in WeKnowIt applications.

2.2.1. Multimodal Interface

From the questionnaire it clearly emerged how in emergency situations people are increasingly using their mobile phone to communicate, due to their immediacy and ease-of-use, although, currently, this is used as a communication medium between family/friends and not with ER services. PCs are the preferred choice for information access as their interface provides the ability to present a greater amount of information.

Interviews with ER team members showed how the possibility of using multiple devices and modalities would be key to allowing real-time information upload and access. For example members of the ER team could adopt touch screen mobile devices, or speech recognisers to communicate during an emergency. WeKnowIt should therefore have interfaces accessible by different devices using multiple modalities both for input and output.

For the first system prototype the following interaction modalities and correspondent devices/platforms need to be taken into consideration:

- GUI, Touchscreen, Pen Input, Keypad (Mobile Phone/PDA)
- GUI, mouse-based interaction (PC)
- Speech Recognition (Mobile Phone, PDA, PC)

WeKnowIt applications must be accessible from all devices but different functionalities and visualisations may be available. For example, speech recognition, where supported by the device, may be used for simple tasks like searching information or tagging content, while it is not needed not be available for more complex tasks (i.e. adding users to a group).

The application will be developed as a web application, supporting most of the current browsers and platforms, with an emphasis on mouse-based or touchscreen interaction.

2.2.2. Content Upload

Uploading content is an important requirement for both ER personnel and citizens as it enables information to be immediately posted from the emergency scene; real-time content upload would also enable direct communication and request of more information (for example, the ER coordinator could ask the Forward Liaison Officer (FLO) to take another image from a different position). The questionnaire also highlighted how citizens are keen to capture content (in particular photos) and share it, primarily with their friends but also publicly on forums. There was a general expression by ER personnel that pictures and videos are particularly useful as they can provide a more easily digestible indication of a situation and avoid the editorial bias of textual communication.

All registered users (or guest users whose account will be created on the fly) must be able to upload

- Images
- Text
- Videos
- Audio

2.2.3. Information Enrichment

Both user-groups expressed the desire to enrich the content they upload with comments, associations and relevant words (tags or annotations). Citizens' primary requirement related to social interactions, e.g. tagging individuals in an image or adding comments. ER personnel were more interested in providing (and accessing) semantic information regarding the content, e.g. type of incident mentioned/represented, incident severity. In addition the geographic location of content was important to both user-groups.

More details on the annotation types will be provided in Section 5.1 where we will illustrate the different types of information enrichment, their role in the system and possible implementations.

2.2.4. Search and Browse functionality

Both user-groups expressed the need to access the information space in a very fast and user-friendly manner. In particular the users want to browse for related information. For example, both ER and citizens want to retrieve geographically co-located information (looking for all the available content near to a chosen area), temporally similar (looking for all the available content in the last 30 minutes) or, for citizens, socially similar (looking for all the available content generated by friends on the Sheffield network).

The search and browse functionalities must be available for all the users (even the non-registered ones).

2.2.5. Personalised Access

Providing a personalised and customised access to information is of much more importance to the organisational user compared to the citizen, as ER organisations need to make sure only the right information is available to the right people.

In order to provide personalised access the user must be provided with the possibility to register. Registration is optional but registered users have access rights to more functionalities. In particular they must be able to choose which modules/applications/functionalities they want to use.

Registered users must be presented with an ability to personalise the system in a way suitable for their user role (i.e. different functionalities will be available for ER users or citizens).

2.2.6. Support for multiple users and visualisations

WeKnowIt must support different visualisations of the content, for example filtering accordingly to user role: an admin ER user will see all the content available, while a normal ER user will see only the content that has been approved by the admin.

Moreover different visualisations should be available to each user, with the possibility to easily switch between several dimensions (i.e. geographic, temporal) to visualise information inside the chosen application.

2.2.7. Information Control

In order to guarantee the usability of WeKnowIt for organisational functionalities, a layer of control over the available content must be provided. For example, content may need to be confirmed by admin ER personnel before being made available to the generic public or some content may be sensitive therefore needing removing by the admin ER

user.

Admin users are informed when new content is added by mail or using automatic alerting features (i.e. RSS feeds).

2.2.8. Communication Prioritisation/Filtering

In order to support the organisational user, WeKnowIt should support prioritisation of communications based on social, environmental and other factors. At times of high network usage, this prioritisation information can be fed into the communications network to ensure key data is received. For example during an emergency the ER user may have sole rights to broadcast information and messages to all other users. It must also be possible to filter and forward information to other control centres or users.

2.2.9. Recommendations

In order to improve the personalisation and increase the usability of the system, WeKnowIt application should provide recommendations to the users. This feature would benefit both organisational and personal users.

For example a citizen when logging in could be recommended to have a look at similar content posted by friends.

2.2.10. Checklist/Task management

In order to satisfy the needs of an organizational user, WeKnowIt applications should provide means to manage checklists or tasks. For example, in an ER context, it should be possible to set up and maintain an emergency response team structure, allocate the work within the emergency response team and defining, describing and assigning tasks to persons.

2.2.11. Feedback/Rating

In order to improve the personalisation and the usability of the system, methods to provide feedback on content recommended or to rate available content should be provided. This feature will also allow distinguishing between organisational and personal intelligence, as while a personal user can rate items of knowledge, the organisational admin will be able to rate knowledge as official approved for distribution.

3. Technical Objectives

3.1. Non-functional Requirements

Table 2 provides a summary of the agreement and divergence between the priorities (High, Medium, Low, or blank if not deemed relevant) given to the non-functional requirements by the two user-groups. Whilst there was a good deal of agreement on functional requirements priorities between the two groups, ER personnel expressed more desire for specific non-functional requirements.

Functional Requirement	Organisational Personnel	Community Citizen
Trust	H	M
Privacy	H	L
Resilience	H	M
Robustness	H	L
Reliability	H	L
Ease of use	M	H
Speed	H	H
Documentation/Help	M	L
Accessibility	M	
Scalability	H	L
Security	H	L
Familiarity	M	M
Latency	M	
Extensibility	L	

Table 2 - Summary of User-group Non-Functional Requirement Priorities

The key non-functional requirements are now analysed into details.

3.1.1. Trust

The users need to trust the system and the information it provides, and this is a key issue particularly for the ER organisation, as they will use the information as basis for their decisions and actions, therefore they need to be able to verify as much as possible its trustworthiness and have a clear indication of this level when analysing it. One of the key ways of assessment of information is independent validation; this might come

from other users or external sources. The system must also always provide means for user confirmation and the responses provided by the system should be consistent and reasoning processes (if applied) should be transparent.

3.1.2. Privacy

Different user roles must be accounted for when displaying information. Moreover all personal information and data should not be visible to other users if the owner does not explicitly agree on that.

The privacy issue becomes even more relevant for ER organisations, as it may be the case of dealing with sensitive information that should not be reported or distributed by unauthorised sources (for example in the case of a person death there must be no unverified rumours or images distributed).

3.1.3. Resilience/Robustness/Reliability/Latency

In order for ER organisations to adopt a new system, this must be resilient, maintaining an acceptable level of operation in case of external influences, such as network failures. For example ER personnel expressed the need for a caching feature, allowing browsing for information offline. In fact the proposed system is intended to "add-value" to the information flow received by the ER organisation, thus in the event of system failure (for example due to internet or mobile network disruption) Situational Awareness should degrade, but not catastrophically.

3.1.4. Ease-of-use/Familiarity

In general for community users the system must be easy-to-use as they will only access it in times of need; however the ER organisation personal are willing to accept that training may be necessary, if the benefits of using the system outweigh the cost of learning to exploit its functionality.

Familiar navigation objects, design icons and similar should be used, to make sure the user instantly recognises the objects in the system and is able to operate them.

3.1.5. Speed

In order to support users while performing their task, especially in emergency situations, the system must be very fast and responsive.

3.1.6. Documentation/Help

A clear and precise documentation and context sensitive help must be provided to the users, especially to the organisational users that will use the more complicated system features.

In order to provide easy interaction in emergency situations, the system must enable quick and non-intrusive help mechanisms, like for example overlapping layers when passing the mouse over an icon.

3.1.7. Accessibility

Ensuring the system accessibility by users with different capabilities and devices is fundamental especially in emergency situations.

Care should be taken in implementing the system conforming to accessibility standards.

3.1.8. Scalability/Extensibility

WeKnowIt applications should be designed and implemented with scalability in mind, to ensure future changes and extensions are possible.

3.1.9. Security

Security is a very important topic, especially for organisational intelligence. Care must be taken in making the online application secure, so that organisational users can safely insert and manage sensitive information.

3.2. Image Sizes, Formats and Compression

The WeKnowIt Data Store will be able to store files up to 500MB, but for the purposes of Emergency Use case, the reasonable limits are:

- 100 MB for video
- 50 MB for audio files
- 10 MB for pictures
- 10 MB for text files (this category contains various file formats - e.g. txt, pdf, ppt, xls)

The system shall be able to dynamically restrict the size of uploaded files, e.g. in case of high user activity, more strict upload limits can be imposed by system administrators on users with(out) certain roles/permissions.

The WeKnowIt system doesn't guarantee to keep the original version of uploaded file. The uploaded file might be converted to more suitable format before being stored (e.g. to a format offering significant size reduction and negligible quality loss at the same time).

There are no restrictions on files formats, but only files of popular formats can be fully analyzed by WeKnowIt tools.

4. System architecture

The system will be created in layered fashion, as explained in Section 4.1. While architecture issues are part of WP6, especially the UI layer is of interest to WP7 and will be discussed in detail. As shown later, the UI will be strictly decoupled which will help to meet the multi modal interface requirement, also in compliance to WP6 approach and progress. The user interfaces (UI) is the end application where all the work done by other WPs is merged. The UI architecture is being designed giving high priority to usability and integration with the services provided by the WP6.

4.1. General Overview

WeKnowIt UI architecture has been designed following a layered decoupled pattern, to help meeting requirements such as multimodality, personalisation, scalability and flexibility. Figure 3 presents a representation of the components involve in the client and server side.

The presentation layer will be created with JSF (Java Server Faces[1]) technology. JSF is a server-side user interface component framework for Java technology-based web applications (it is a part of Java EE technology stack). It helps to develop web applications by providing support for typical tasks (like handling of events and user session, data validation, internationalization etc.). It also offers a library of custom tags (JSP tags) that can be used to express UI components within a JSP page and for wiring components to server-side object through its bean facility.

In order to provide rich interactive web applications, which improve user experience, the AJAX approach will be adopted. In order to facilitate the integration of AJAX into the JSF the use of jMaki Framework[2] is proposed.

jMaki is a lightweight client/server framework, which creates JavaScript applications that communicate to various server-technologies, in particular JSP/JSF. jMaki acts as a java wrapper of widgets built from different Ajax frameworks (including Dojo[3], YUI[4], prototype, etc). Each wrapped component can be included in JSP or JSF tags. All the components provide client services such as publish/subscribe events between components, JavaScript action handlers, and a generic proxy to interact external services.

[1] http://java.sun.com/javaee/javaserverfaces/

[2] https://ajax.dev.java.net/ -

[3] http://www.dojotoolkit.org

[4] http://developer.yahoo.com/yui

The use of jMaki facilitates as well the communication of Ajax widgets made in different frameworks (for instance, dojo widgets communicating with YUI widgets).

The jMaki project allows using a JSF managed bean to populate the data of a widget wrapped as a JSF component, much the same way as data in any other JSF component would be populated.

Since the UI most be integrated with the services provided by WP6 (main architecture) through web services, it could be easy to call just the existing web services and provide directly the answer in the UI, however the UI comprises of more complex functionalities, by letting the user interact with the interface in an event-driven model (see subsection "Example use cases"). This helps the user to compose the type of services he needs by combining the different components within the interface (see Sparks framework). The complexity in the UI introduces a thin presentation logic layer in the server side. The presentation logic layer (adaptor) will help to integrate the services provided by the UI with those exposed by WP6 through the use of web services.

The presentation logic layer is responsible of preparing the data obtained in the UI in terms of the parameters expected by the web service and the other way around, this is, when a response is obtained from the web service, the presentation logic prepares the data to be passed to the components within the UI.

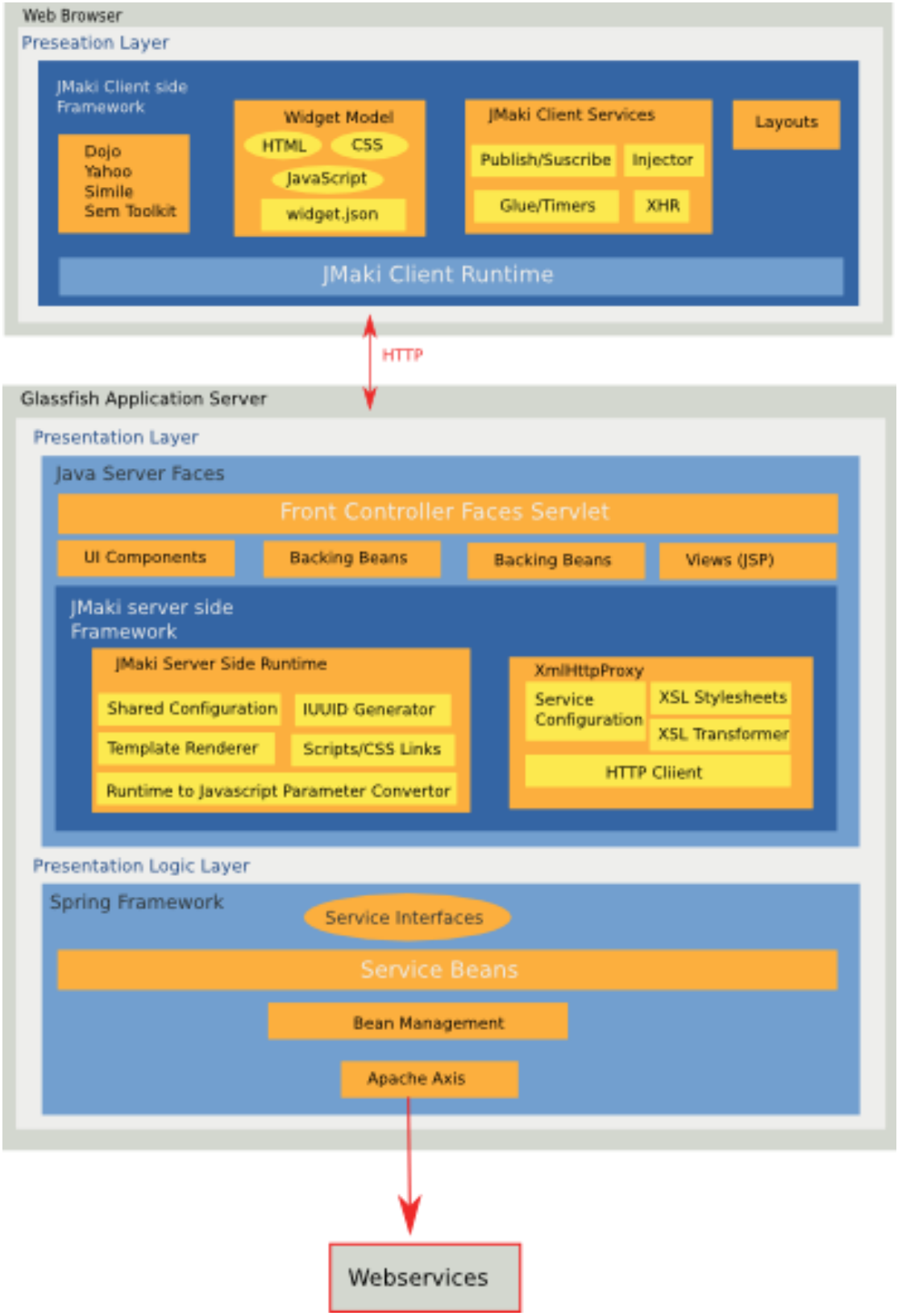


Figure 3 - UI Architecture Overview

The presentation logic layer relies on the Spring framework, which helps in managing business objects and their dependencies. Through inversion

of control, Spring allows to be loosely coupled with the elements involved in the business logic.

Spring is composed of various modules; in particular Spring Context and Spring Core will be used to configure the application through JavaBeans.

Through different Service beans will act as a façade offering the functionality provided by WP6. This functionality will be requested by consuming the web-services exposed by WP6.

4.2. JavaScript Libraries

As shown in Figure 3 the presentation layer (client-side) renders different type of widgets (components) coming from different Ajax Frameworks including: Dojo, YUI, Simile[5].

The use of these widgets will facilitate the integration of external services (for instance the use of yahoo maps through the YUI geocoder widget).

However in order to provide personalised visualisation according to the user's needs a customised UI referred to as Sparks Framework is proposed. Sparks Framework will help to relate annotation facilities with graphical components.

4.3. Sparks framework

This framework will provide a data structure for supporting the incorporation of semantic annotations into graphical objects, called semantic-aware widgets or components. This framework will help in the flow of information and the automatic reuse of annotations. For instance dragging and dropping an image from a carrousel into a particular event component, would enable the automatic annotation of the image with information obtained from the event.

All the components (including those from external frameworks (i.e. YUI, Dojo) will be created or wrapped with the Sparks framework. In this way the UI will be composed of semantic-aware graphical components.

Each component will be capable of establishing semantic relationships between them. Each component will have an RDF feature that will relate it with a concept. They will be capable of self-manage the event (triggering and consumption), access, and validation control.

The components will contain a data structure, which considers the following attributes:

- Composition with other components
- Permissions (edit, view, remove)
- RDF concept

[5] http://simile.mit.edu/

- Publish (Events)
- Subscribe (Events)
- Position within the page container

More specific components will extend the semantic task component data structure. For instance a Carrousel Widget will be composed of a set of Image Semantic widgets, when dragging an dropping an Image the control within the component will be able to identify the permitted relationships (where it can be dropped). Components will be capable of maintain semantic relationships when linked to other components.

4.3.1. Example Use Cases

In the following sections some example use cases will be presented, that exemplify how the architecture model will comply with possible scenarios and user requirements.

Adding a picture to a fire event

Figure 4 presents the UI for a fire event. Each element is a semantic-aware component. Consider the case in which a user adds pictures from its carrousel into a fire event. Each picture from the user's carrousel contains annotations including both manually (location, name, etc.) and automatically (author, location, etc.) added annotations.

Considering the following annotations for a picture: name, geo-location, location (within the UI), event, and author. When the user uploads the picture he can automatically edit these annotations, however it would be very likely that the user does not have information related to the geo-location (in case this is not supported by the device), in which case the annotation would remain empty. When trying to relate the uploaded picture with the event, he would be able to relate the picture with the position from where the picture was taken (finding for instance the street on the map). When dragging and dropping the picture into the semantic-aware map component the geo-location annotation would be automatically updated.

The users can instantiate a new concept by dragging it to the map. For instance, when the new picture concept is dragged from the carousel to the map an upload picture information box appear. Then, the uploaded picture can be annotated using the previous drag and drop facility.

Figure 4 - Event-based User Interface

Linking concepts

By means of the Sparks framework, the user will be able to link components which at the same time will permit the linking of concepts (see Figure 5).

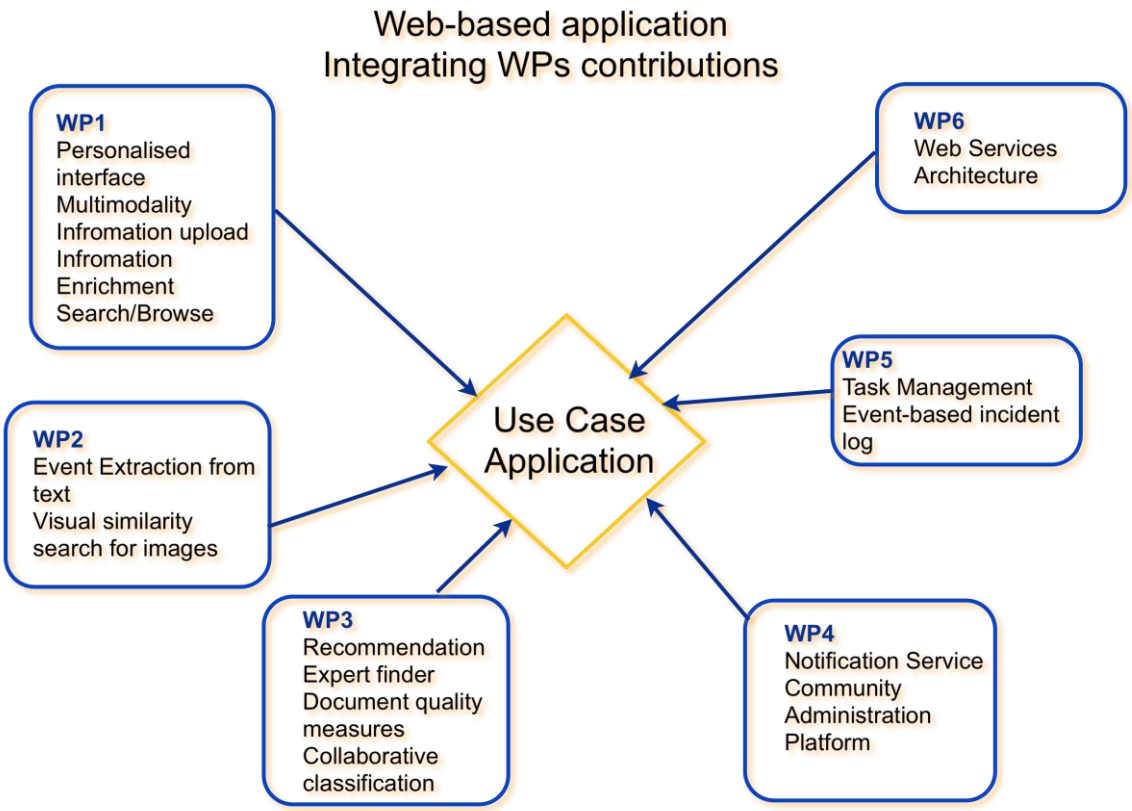
Since the user should be able to add or remove Semantic widgets from its main page, the system should be able to keep track of the changes the user performs to customised the interface. This is, the presentation layer will keep a record of this information.

Figure 5 - Fire Event Presentation

5. Backend Processes / Systems

In the next sections, short descriptions of all Intelligent Layers are presented. The provided technologies will be linked to the use cases.

In Figure 6 a diagram of the main technologies provided by the WPs to WP7 are illustrated.



16

Figure 6 - WPs Contributions to WP7

5.1. Personal Intelligence

Goal of personal intelligence is to develop knowledge management methodology and technologies for individuals, providing support for knowledge upload, access and sharing. Whilst the generic methodologies and tools for Personal Intelligence are interest of WP1, they will be now discussed in details to explain how they will be customised in WP7 for WeKnowIt use cases, in this instance the Emergency Response.

The methodologies and tools presented in the next sections have been devised for Personal Intelligence, but then customised to support also an organisational user, thus emphasising the convergence between Personal

and Organisational Intelligence in the effort to build Collective Intelligence.

5.1.1. User model

User modelling is fundamental to meet the high priority requirements for both user groups, as it enables login functionalities and personalised access to the system. As expressed in the scenarios identified in D7.1, ER personnel and users have different needs, for example an ER Admin user will be able to edit, modify, approve or reject content, functionalities that will not be available for a normal citizens. In order to achieve this, storing user details is fundamental: in particular this will allow to recognise what type of user is logged in the system, what functionalities will be available, how the interface is presented. Detailed examples of how storing and modelling user details can contribute to personalised interfaces and services are shown in Sections 7.3.1, 7.4.2 and 7.5.3.

The methodology adopted for storing and modelling user details in WeKnowIt involves the use of semantic and the implementation of an OpenID server that fully supports all the OpenID 2.0 specifications.

The OpenID 2.0 specifications propose a method for storing key/values attributes (personal data), in which the attributes types are defined with the RDF syntax. AsterID [8] is the WeKnowIt proposed integration and extension of OpenID standard to support the storage and retrieval of user details and information and to support the communication between users.

5.1.2. Information enrichment

All the information uploaded to the WeKnowIt Data Storage can be enriched using different methodologies:

- Tags
- Ontology-based annotations

Information Enrichment is necessary in order to structure and organise the content that is uploaded by users, as it will provide metadata that can be used later on for retrieving and sharing the content: both methodologies envisaged for Information enrichment have been researched by WP1 and WP5 as they involve both Personal and Organisational Intelligence.

For example, when a ER staff member is sent to the emergency scene, it must be provided with not only ways to upload photos about the incident but also with means to report exactly what the incident is, what are the consequences and so on. More details can always be added at a later time.

Information Enrichment methodologies will also closely interact with WP2, as they will provide the basic annotations for the Media Analysis module to

work on. For example Ontology-based annotations, being very precise, may form the basis for algorithms that will learn from user annotation to provide recommendations and targeted content.

Tagging

WeKnowIt will adopt Web2.0 tagging as interaction paradigm for quickly adding tags to objects that are uploaded or viewed.

State of the art work in automatically processing user created tags and matching them to one or more ontologies will be adopted and improved, so to maximise the utility of the free user annotations.

Moreover tags added by the user will be used as a tag cloud for browsing the information space and refining queries.

Ontology-based annotation

In order to have very precise annotations (of instances linked to ontology concepts) without burdening the user with complicated and time consuming interaction, in the first prototype ontology-based annotations will be achieved by providing a form that users can fill when uploading new content. This methodology, devised in WP1, has been extended for WP5 as it enable more sophisticated knowledge management, necessary for Organisational intelligence.

This proposed methodology [9] will enable administrator users (for example admin ER personnel) to create a form with pre-defined questions to ask the user when submitting content using a graphical interface: an intuitive form creation process helps users to logically divide forms into reusable conceptual sections and fields. For example a section on "Personal Details" (e.g. forename, surname, address, etc.) can be largely reused elsewhere. When creating a form, users are suggested potential forms and sections to be reused according to the current context. Users can also proactively search for reusable items and forms.
Forms are internally represented as OWL classes (<FormName> Class) with their own NameSpace; form sections are represented as OWL classes (<ConceptName> Class), which can have subsections (related classes) or individual fields (properties). Relations among concepts are established and represented as OWL relations. Each form is stored as OWL ontology in a queryable store so to enable reuse of concepts (i.e. sections). When parts of forms are reused, the underlying part of ontology is shared among the forms (or interlinked with OWL-equivalent relations), hence creating a web of interconnected ontologies. A meta-mapping ontology is automatically created and maintained, to store all the interconnections.
Form filling generates RDF triples that are stored in an RDF store referring to the OWL ontology. Validation rules ensure the filled values conform to the OWL requirements. Forms are filled by either selecting pre-defined values e.g. via drop down menus (hence generating URIs) or by inserting

snippets of texts or images. When snippets are provided, they are annotated with the semantic type of the field they fill.

In the first prototype two forms will be predefined for two classes of users: ER personnel and citizens, but these forms could be changed at any time by an admin user. For example, when a new emergency happens an Admin ER Professional may want to insert a new field in the form to ask for specific details, or a professional for another organisation (i.e. fire services) may want to insert a form representing their view of the situation. This methodology has the advantage of supporting the definition of communities' views within or across organisations.

The information requested to a citizen will be simpler than the information requested to an ER professional.

For example when uploading new content a citizen will be asked:

- What type of emergency is it
    - Fire
    - Flood
    - Road accident
    - Explosion
    - Other
- Are there any casualties
    - Yes
    - No
    - Not sure
- Location (if not automatically gathered)
- Any other information

An ER professional user will be asked:

- What type of emergency is it
    - Severe Weather
        - Storms
        - Flooding
        - Gales
        - Snow
        - High and low temperatures
        - Water shortages

- Human Health
- Terrorism
- Transport Accidents
  - Train crash
  - Road crash
  - Ferry accident
  - Aviation accident
  - Rail accident
- Fire
  - Residential
  - Industrial
  - Other
- Animal and Plant Diseases
  - Foot and mouth disease
  - Rabies
  - Avian influenza
  - Other
- Public Protest
  - Nationally co-ordinated protests and large localised protests
  - Industrial action
- International Events
- Industrial Technical Failure
  - Gas
  - Water
  - Electricity
  - Telecommunications.
- Structural Failure
  - Land movements
  - Structural failure in the built environment.
- Chemical, Biological, Radiological and Nuclear (CBRN)
- Industrial Accidents and Environmental Pollution
  - Accidental chemical release

        - Accidental biological release
        - Accidental radioactive release
        - Major industrial fire
        - Accidental explosion
        - Land or water environmental contamination
    - Other
- Location (free-text fields)
    - Location of emergency
    - Access
    - Egress
    - Rendezvous point (RVP)
- Hazards (yes/no) plus a free-text box
- Are there any casualties
    - Yes
    - No
    - Not sure

Forms can be created accordingly to the type of media submitted, the type of user (professional, citizen) and the situation in which the user is.

5.1.3. Search

In the first prototype of the ER application the user will be able to exploit the semantic behind the collective knowledge when searching. All the user types need a search feature, as it will empower them to effectively access only the needed information in real-time.

Semantic search will be available as well as simple keyword search, providing multiple search possibilities:

- By content type
    - Image
    - Video
    - Text
- By similarity
    - E.g. More images of fire
- By location
    - Images from similar location

Both graphical and standard search interfaces will be provided to the user, accordingly to the type of search performed. For example when searching by location a graphical interface would be ideal, as it would allow the user to expand and reduce the location using the map displayed.

A multi-perspectives interface will be implemented for the visualisation of the results.

5.1.4. Browse

As the amount of content that will be uploaded to the WeKnowIt data Storage may be quite extensive, all type of users must be provided with means to navigate this information space in effective ways. Especially during an emergency users will post many photos, videos and text and it will be very difficult for a user that needs to take important fast decisions (like an ER Admin user) to orientate and make sense of the large amount of data. Faceted browsing techniques will therefore be adopted for helping the user navigate the information space, with some pre-defined dimensions that have been defined during the user studies.

Amongst the dimensions that will be used to display information:

- Time
- Type of information
  - Video
  - Picture
  - Text
- User

5.1.5. Access

In order to guarantee the uptake of the system in Emergency Response situations, it is important to pay attention to the security and privacy issues, finding ways to enable fast and secure access to possibly sensitive information.

As OpenID 2.0 is the upcoming standard for storing digital identities, using XRI for identifying users, the ER application will use OpenID to authenticate the user and store details.

Since the personal intelligence is highly correlated to the user's identity, it has been proposed that the WeKnowIt system serves as an identity provider (WeKnowIt OpenID). OpenID models the user's profile and represents the user's identity to other services. Being an identity provider will enable the extension of the user's representation. This extension will model the user identity considering for instance user's historical information, providing in this way tendency perspectives in the user's behaviour.

In this architecture it is considered to maintain a triple store of the user's identity representation, separated from the triple store where the WeKnowIt, social, mass, and organizational intelligence will be stored. The user's personal information will be addressed in the WeKnowIt main triple store through the user's OpenID URI pointing to the WeKnowIt OpenID server.

User Registration

The user will be able to register their details during the sign up procedure or they will be able to submit content without being registered. In this case the system will create an account on-the-fly and the user will then be able to complete the registration with all the details at a later time.

This is to ensure that everyone in an emergency situation can submit content even if not part already of the community.

5.2. Media Intelligence

Media Intelligence aims at the development of intelligent, automated content analysis techniques for different media to extract knowledge from the content itself. Within the WeKnowIt project three types of user-generated media are considered for analysis, namely text, visual and speech and will be presented shortly in the following subsections

5.2.1. Text Analysis

Text Analysis attempts to explicitly extract events, which can be constructed by the users (e.g. ER Organisation event logs) but they are also implicit in the community social network data. This requires the use of Information Extraction, Event Type Identification and Event Identification techniques. The state of the art has mostly dealt with large, static, well-edited, coherent and complete text, however social data is Ill-formed, ungrammatical, colloquial, short text of interleaved "conversations".

This requires that Text Analysis must consider the context of the information in order for it to be disambiguated and validated. Location gazetteers are constructed using the available location specific resources from postal and map services, and from general services (e.g. Google Maps™, Openstreetmap). The information extraction system uses these gazetteers and pattern matching to extract temporal and location entities. Where unrecognised terms are found error correction attempts to resolve these considering the similarity between the unknown terms are the gazetteer terms. Ambiguous entities are resolved by considering their context: i.e. using other entities: in the text, by the text's author, in similar texts (i.e. similar in time, content). The social network of the text's author can be used as part of the context, exploiting the Social Intelligence work in WP4.

Topics are constructed by considering removing location, temporal and user information from the text, then extracting the keywords (also called terminology extraction) these terms are then clustering (using hierarchical clustering) to provide a set of potential topics. The texts relating to given locations and topics are then clustered (using hierarchical clustering) into events.

In terms of the ER use case this analysis will provide the ability to extract event information from public forum conversational data, such as that found on SheffieldForum concerning the June 2007 Flooding in the Sheffield region. The results of the analysis hope to show that how explicitly extracting the collective intelligence, which is implicitly contained within such forums, can provide information capable of meeting the requirements of the ER user groups, providing input into the Organisational and Personnel Intelligence Layers.

5.2.2. Visual Analysis

The work on Visual Analysis with respect to both the emergency response and consumers' social group case study includes research on determining whether digital still images relate to the same object and whether specific image scenes may be identified based on the actual content itself. This involves overcoming typical research problems like spatial/temporal decomposition and structuring, object/event detection, recognition and tracking and the investigation of appropriate visual features, metrics and supervised learning approaches using annotated training data.

Developed technologies apply with equal force to both WeKnowIt case studies. For instance, within the consumers' social group case study, visual retrieval and localization of multimedia content together with advanced text processing tools will enable WeKnowIt users to efficiently search and retrieve information about specific landmarks or points of interest during their travel, as well as facilitate the way they share their personal multimedia content acquired during their trip. When dealing with the emergency response case study, raw multimedia content (e.g. still images or audio fragments acquired by digital cameras or mobile phones) uploaded directly from emergency event sites will be analyzed and its information exploited towards efficient, automated and quick identification of objects, landmarks or events of interest.

From the technical point of view, using a large database of geo-tagged images acquired from Flickr™ as well as the Sheffield City Council data, the work within WeKnowIt matches a given query and returns a ranked list of images according to visual similarity. The geo-tags of the returned images are used to provide an estimate of the location of the query photo and display it using Google Maps™. To achieve this goal a number of technologies is applied in order to perform efficient object detection, where those objects are under variable scales & poses, using feature points (SIFT/ SURF), Harris- & Hessian-affine detectors and optimized

RANSAC inliers to ensure geometric consistency. This process creates a visual vocabulary of points for each image under consideration, which can be used to define comparable visual elements and thus provide the means to compare both isolated objects and entire scenes.

In addition, there is also work on analyzing tagging patterns and combining them with the above content feature extraction methods, generating, thus, intelligence from multimedia social tagging systems. More specifically, emphasis is placed on using all available "tracks" of knowledge, namely social knowledge (i.e. knowledge that can be derived from tagging systems, e.g. tag co-occurrence), semantic knowledge (i.e. knowledge about the meaning of the concepts, like for example hierarchical relations among them), and content-based knowledge (i.e. the low-level features of the multimedia data). This way, exploitation of users' annotations, such as those provided by the Personnel Intelligence Layer when uploading images of incidents, produces semantic metadata and provides added-value to the available multimedia content.

5.2.3. Speech Analysis

The key focus of the speech analysis work in WeKnowIt is to enable Out of Vocabulary (OOV) words detection to facilitate Phonetic Search techniques (i.e. searching for words not present in the vocabulary, e.g. people, places, and company names).

This is performed using sub-word term candidate rescoring in word lattices produced by optimized Large Vocabulary Conversational Speech Recognition (LVCSR) algorithms, which employ an innovative new method that combines word and sub-word decoding.

For an incident on the scale of the Sheffield 2007 floods there are likely to be tens of thousands of calls logged by the ER organisations. Speech analysis of these logs can provide a means for these logs to be searched to locate potentially important information.

5.3. Mass Intelligence

Goal of the mass intelligence is to provide methodology and appropriate services that allow creating a more structured and aggregated view of incoming information in the WeKnowIt system.

Mass intelligence services concentrate on areas of information enrichment, information search, information recommendation according to predefined or ongoing tasks and providing data for event alerts defined in the system. Mass intelligence will provide services in the areas of classification of information in the system, recommendations, information tracking with support of ER tasks or events, data aggregation and trend analysis. The main consumers of mass intelligence services are coordinators and members of ER teams.

5.3.1. Categorization and clustering

ER team expects to receive aggregated and already structured information from the system. On the other hand, incoming information can be massive, using various media and carrying different semantic value.

Information enrichment and categorization is often a prerequisite for other offered services. Incoming information are analysed by media intelligence components, enriched with spatio-temporal information as well as categories assigned both by users and system. ER personnel need to have accurate and up-to-date overview of the situation that includes important and sufficient facts, but not overwhelming with detail level. Mass categorization and applied information clustering will help to extract the important parts of content and establish certain semantics that could be used in further services consuming such data. Information enrichment can include detected type of event (e.g. flood, fire), actors involved, places as well as fit into certain informative groups. Such categorization and clustering of information will also help to overcome the possible information overload in case when too many similar or repeating messages are coming to the system in a relatively short time.

5.3.2. Trend analysis and predictions

Enriched and categorized information will be used for trend analysis to create a temporal and spatial model of developing emergency. When emergency situation occurs, citizens start to contact different emergency services and teams. Different kind of information is coming to the system, as people call, send SMS or photos. Mass intelligence will provide tools to detect such localized information bursts, which can suggest emerging emergency event or spread of emergency situation into new places. In case WeKnowIt system will have access to geographical or weather information, provided services in mass intelligence will be able to use predefined scenario templates to predict potential development of the emergency situation.

5.3.3. Recommender systems

In addition to event prediction, system could utilize information from defined scenarios to isolate outlier events or provide suggestions about information which may be missing in the system.

Outlier detection algorithms used in recommendation systems could suggest events that are weakly connected to the major emergency, geographically belong to other events or are effects of existing situation, but require different handling. Information about missing data in the system may be crucial for coordination of emergency actions – it could suggest potentially endangered locations currently not appearing on event map, measurements that could play vital role in estimating importance of such events.

Another task of recommendation system in supporting emergency response case is providing users with annotation recommendation to uploaded resources in easy, quick and intuitive way. During emergency citizens as well as ER team are under stress and time pressure. There is no time to provide additional annotations, which have significant value to the ER team, as they are used for making important decisions. One of the examples is providing appropriate tags based on uploaded content, user location and current information about emergency events. Recommendation is composed using description of emergency scenario together with inputs already present from multiple users (from current as well as previous incidents). Such service will utilize results of visual analysis provided by media intelligence components.

5.4. Social Intelligence

5.4.1. Community Design Language

The Community Design Language (CDL) will be a formal language for storing and checking access rights against a facts base. The CDL is built such that access rights can be formulated in a very general way, allowing the storage of a huge set of different access right types.

The Community Administration Platform (CAP) supports organizational and social roles. A organizational role is linked to a position in an organization, e.g. a company (job position). A job position has material and immaterial components. The immaterial components are sub-structured in tasks, targets, authorizations and responsibility [10]. The CAP supports the mapping of authorizations to organizational positions / roles. A social role we define as quantity of expectations of and towards a member of a community. These expectations include duties and rights. Such rights can be bound to social roles represented in the CAP.

A overview of all components of the Community Administration Platform (CAP) is given below. The picture is a free interpretation of [11].

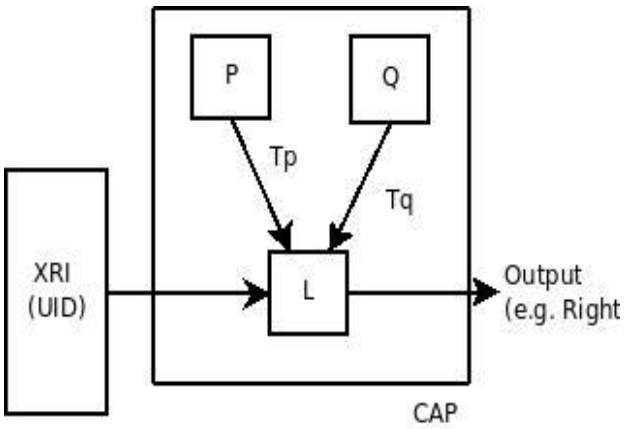


Figure 7 - Components of the Community Administration Platform

As illustrated in figure 1 the CAP consists of three components. The logic part P implements the authorization decision-making depending on its input parameters. CDL will support different authorization conditions (e.g. users, permissions, objects, timeframes, devices) being able to be checked, if access should be granted. The elements of the authorization conditions can be assigned to lattice structure elements, allowing them to be organised.

The policy component P allows the definition of policies. We define a policy as a specification of rules and/or conditions necessary to grant access to a resource. The transformation Tp ensures, that policy P is expressed in a way, L can interpret it. Actually, Tp is the CDL representation of the policy.

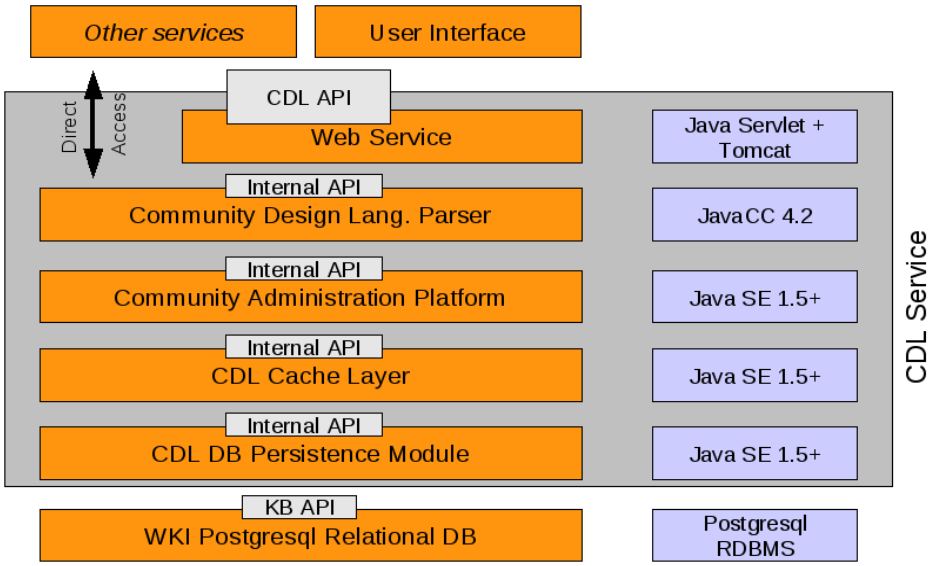
It is important to notice, that unique Ids identifying all elements if authorizations conditions must be provided externally. Unique Ids therefore a necessary for users, objects, in case timeframes and devices. CDL/CAP will not provide them!

The authorization decision takes as another input a query Q representing a specific authorization request. Again, Tq is the formal representation of Q in a language understood by L, the CDL.

The CDL as formal language of L enables any system to represent its rights sets, store access rights in an efficient and generalized way and check before executing commands against the facts base. Self-organization is supported by the ability to delegate access rights through representing access right delegation as a rights set itself. By this, the CDL is a potential, common infrastructure component for all WeKnowIt services.

A technical overview is given in the diagram below.

T4.2 CDL Technical Overview



The facts base will be stored in the WeKnowIt data storage and accessed via standard APIs. Postgres is preferred due to its scalability, transaction safety and free availability, but can be substituted by any comparable component offered by WP6.

The CDL Persistence Module is an abstraction layer enabling data retrieval and storage with the database. It will be realized as a library in Java Standard Edition. The usage of Java Beans and other frameworks have been evaluated. The decision was made, to implement an own persistence layer. The main reasons for this decision has been to be able to optimize database access for our special needs, as we expect performance as a main issue concerning access rights. And the easiness of implementation, not making it necessary to implement in huge and complex frameworks as Java EE.

Seeing performance as major issue concerning the CDL implementation, a cache layer will be implemented. Caching algorithms – e.g. used in OS implementations – will help, to minimize costly database accesses while ensuring a constant, not to greedy memory usage for this component. Abstracting a cache layer will help us to be able to implement different, optimized caching algorithms and maintain them separately.

The Community Administration Platform layer is one of the two core components of the CDL. In this layer, the core algorithms will be implemented. The CDL will support an (theoretically) unlimited number of access conditions (e.g. user, permission, object, event, device, timeframe). All elements of such sets (e.g. users) can be assigned to lattice structures, enabling explicit attribute assignment. Implicit attribute assignment is also an issue, we plan to implement. Attribute elements can be combined to contracts, representing complex and efficiently to

formulate access rights. Our main effort will be, to ensure performance, flexibility and scalability.

As the CDL will be a formal, structured language (e.g. like SQL) a parser layer will be implemented. Currently we plan to use the Java Parser Generator JavaCC. This widely spread Java standard component will allow us, to minimize our own efforts in this field. The CDL itself forms an interface to the outside of our component, as all functionality can be used while formulating commands in the CDL formal language.

In order to work in a service oriented architecture (SOA) we plan to implement another layer, encapsulating the CDL functionality as a webservice. We plan to be able to receive messages as a service (of course, internally formulated in CDL) and answer those the same way. Of course, other services using the CDL component will use the API of this layer.

The last layer is optional and can either be a graphical user interface or another service. This component is loosely coupled, as it is a complete self-containing component. It is possible, if supported by the architecture, to bypass the logical webservice layer and contact directly the CDL parser.

5.4.2. ER Use Case applications

Retrieve team memberships

CDL supports organizational and/or social roles, allowing it to retrieve all members of such a role. We can use this for example, to retrieve all members of the gold team. WP5 might then use this list to inform them electronically about a major emergency case.

This support applies e.g. for step 4-6 to the ER case, allowing to define organizational roles for gold, silver an bronze level. Step 15, receiving personal information after a login, can be supported by retrieving all organizational roles of John. Step 20 is supported by providing all registered (active) Unique Identifers (UIDs).

As a benefit for the ER team, we see a faster team notification and a better response time for the rescue services. Alerting can be done automatically and quasi-parallel.

Formally, in the CDL one can assign unique user UID provided by WP1 to lattice structure elements with component P. Component Q then can ask for lattice structure assignment and retrieve the list of assigned UID, e.g. as a vector or list.

Assign team memberships

CDL can assign users to organizational roles. Step 7 of the ER use case can be supported by a service, which assigns John to the silver strategy team.

Formally CDL can assign a UID to a lattice structure element.

Assign access temporary right

In step 10 ER use case, Mark receives an SMS in order to upload a flood picture to the command centre. CDL can support this by creating a policy, allowing Mark to upload this picture for the next 10 minutes via his mobile phone. After this, the access right is no longer valid. This is an important restriction: Probably Mark shall not be able to upload pictures forever, but only during a specific time period. For the sake of efficiency, the limitation is compiled into his access right. It is not necessary to remove this access right later manually or by a batch job.

By this, benefits for the administrative overhead can be minimized: The ER staff gets better information by the uploaded pictures while not having to deal with administrative overhead, who and when somebody is allowed to upload resources.

The same way, step 18 can be supported, by assigning upload rights to Andrea for some specific time.

Technically CDL must receive a UID – probably a temporal one, created on the fly – and assigns a role to it, allowing the upload of pictures (object condition) via mobile phones (device condition) for the next 10 minutes (timeframe condition).

Assign sets of resources to public

In step 19 John wants to publish a set of flood photos to anyone via the web interface. CDL can support this by offering a service allowing John to give read access to the complete set of pictures. John is acting here in his role as EPT. He can also delegate this access rights to another role, e.g. PH or press liaison.

For John it is a very easy, flexible and efficient way: He can give the public access to such pictures with one CDL transaction. If John decides to delegate his rights to EPT, the rights delegation is also a very easy way. It is important to note, that John can only delegate the right to grant access to those pictures to the public to EPT. He is not forced to delegate all his authorizations. By this, EPT has limited exactly this authorization (of course, EPT has access rights through their own roles).

Doing so, reaction times and the time to inform the public with relevant information can be significantly reduced. Through delegation, the process of sharing pictures/resources can be shared, leading to a reduction of work for key personnel.

Technically all pictures must have been assigned to a subset of a lattice structure. Via the service, this structure is assigned with read access (permission condition), via web interface (channel condition) to all public (user condition). Perhaps in a certain, limited quality (e.g. <=1024x800px, quality condition).

Retrieve relative (social) roles

CAP will support relative roles. In contrast to a role "firefighter" or "system administrator", a relative role is dependent on a parameter.

Relative social roles required by the ER scenario for the users of the type citizen are:

- Family
- Neighbours
- Friends

The CAP platform will provide these roles as relative social role type parameterised by the user in order to keep the number of specific roles down. For example, family(i) denotes all the family members of user i.

Emergency Notification Service

Finding out about the status of family members or friends makes up to 59% resp. 29% of all communication (see D7.1, p. 167) during an emergency. If we can reduce this traffic significantly by offering a emergency notification service allowing to share ones personal status with your family and friends, this will lead to several benefits.

Such a service will first of all reduce communication and status requests to emergency recovery personnel, giving them more time to deal with real recovery actions. The chances of overloads of the communication networks can also be reduced. Third, for the citizens, this service can minimize the psychological stress time not knowing about their family status. Also, the communication time and effort for the family members can be reduced, as bilateral communication can be avoided by a hub oriented messaging service.

In an article in Nature[6], Winerman stresses also the importance of internet technology to improve communication in crisis situations. She expects technologies being developed to enable this during the next 5-10 years. This notification service can be a first step in this direction.

The specific details and communication structure will have to been refined during the project. A very straightforward and simple model consists of a web service: Any pre-registered citizen can sent a status request message to this service. By sending this, also a status for himself is sent and stored at the service. Any family member sending such a request as well can receive the pre-sent statuses of the family, giving them information.

The following sequence diagram shows the communication flow.

[6] Winerman, L. "Crisis Communication". Nature, Vol. 457, 2009

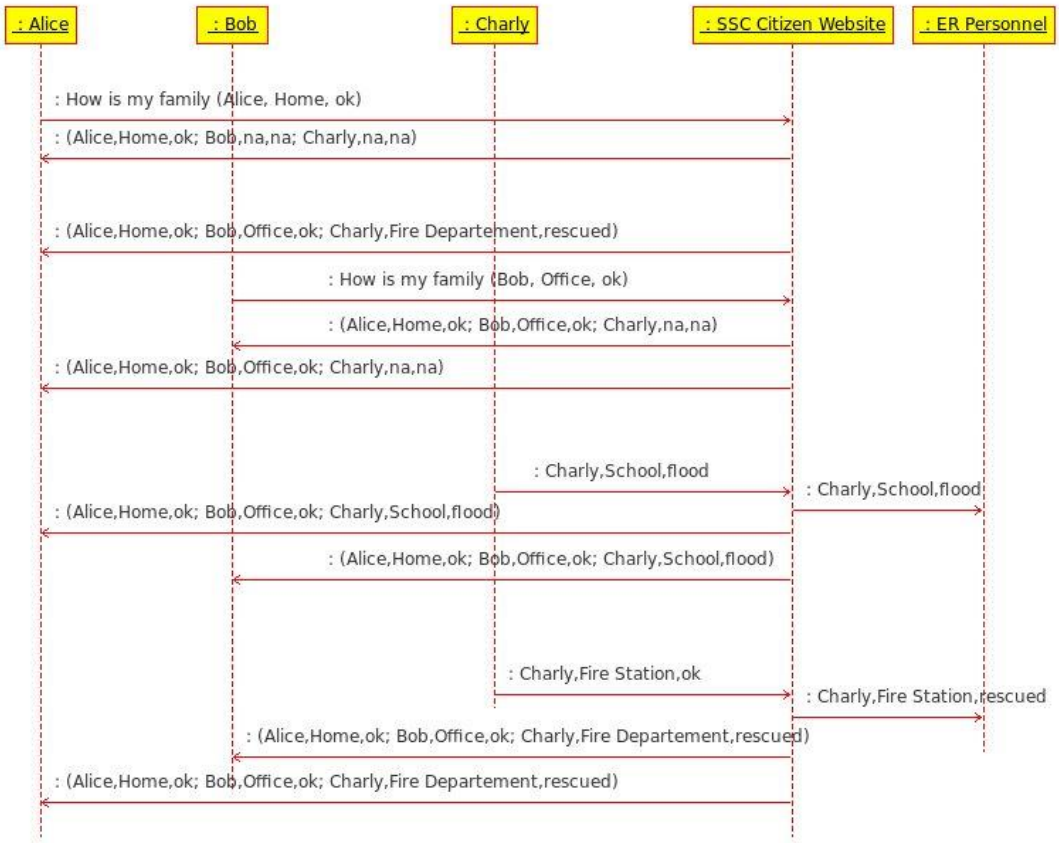


Figure 8 - Sequence Diagram of the Emergency Notification Service

Let's assume a family of mother Alice, father Bob and child Charlie. All three have registered with the SCC emergency web service and taken the role of family members co-mutually. Alice gets news about an emergency and sends a request "how is my family" to the service, together with her status (ok) and her location (at home). As neither Bob nor Charlie's status is clear, she receives a message, stating, the current situation of Bob and Charlie is unknown.

Bob sends the same request together with his status (at office, ok) a few minutes later and receives the answer, that Alice is fine at home and Charlie's status is unknown. A push message is sent immediately afterwards to Alice, informing her about the status change of Bob.

Charlie got stuck in the flood in his school and informs the SCC service about his dangerous situation. Alice and Bob are informed immediately. Rescue services as well get word about Charlie's status, helping them to identify people in dangerous situations. Rescue personnel then rescue Charlie and bring him to the fire department. Charlie sends an update message to the SCC website. Automatically, Alice, Bob and ER personnel are informed about his new location and new status (ok).

For the other relative social roles, a user may opt-in for the emergency notification service.

The access policy for family(i) for the ER scenario informally specifies:

- Each family member of user i has all the rights of the social roles of family(i).
- Each family member may use the emergency notification service.
- ER-personnel may us the location service of the emergency notification service.

This service can be further improved by combination with other WP services. WP1 can contribute the Personal Intelligence Layer, by providing users the capability of setting a status, sending a message and receiving notification messages on multimodal interfaces. For example, during an emergency a citizen may send a status message using WeKnowIt mobile interface: as their family is not logged into WeKnowIt but their contact numbers are available in the registered user details, WeKnowIt may send the notification message using a SMS gateway. At a more technical level WP1 will contribute to the Emergency Notification Service with services for storing and retrieving user details and for sending and receiving status messages.

This service, organically thought for benefitting Personal Users, can be extended to more Intelligent Layers, thus enhancing the multi-way communication protocol during an emergency.

For example, the same service may be adopted inside an Organisation for exchanging messages between members of a team. Moreover also Organisational to Personal Intelligence communication can be achieved using this model: when an event starts, personal users can subscribe to the event to be alerted by ER team members about the status of the event. By using this service the notification would be immediate and effective.

5.4.3. Community Analysis Tool (CAT)

The community analysis tool will provide social network analysis (SNA) methods to WeKnowIt. The central components of the CAT are the algorithms providing the SNA. Additionally, data structures to store the graphs for the analysis are provided.

One objective of the WeKnowIt system architecture is to guarantee the independence and reusability of all intelligence layer modules. Consequently, the CAT may not have direct links to data sources. Therefore, it has to be the responsibility of the applications making use of CAT modules to input data. Direct links between the WeKnowIt data store

and WeKnowIt intelligence layers exists only for those data a specific WeKnowIt intelligence layer owns. Therefore, the graph data structures provide basic interfaces to input vertex and edge data. It is the applications responsibility to gather required data from the WeKnowIt Data Storage and to create and update graphs using the provided interfaces. The community analysis tool is not the owner of any data but operates on copies of the data it receives. An application should not rely on receiving graph data from the community analysis tool but have the original data in the WeKnowIt data store. In case of system errors it might be necessary to recreate all graphs.

The community analysis tool consists of four layers. The graph layer provides data structures for directed, undirected simple graphs and bipartite graphs for affiliation networks. All graphs can either be weighted or unweighted. The analysis layer provides algorithms to analyse the data structures of the graph layer. A caching layer can be used to cache analysis results that are expensive to calculate. The caching layer can be left out if not needed (e.g. graph size is small). The fourth layer is optional, too, and provides the semantic of the analysis results if the semantics of network and it's context has been made available to the CAT by the calling system component.

The community analysis tool will provide a set of SNA measures useful for the analysis of online communities. We differentiate analysis methods by their view on the network. Actor level methods look at all actors separately and assign values to them e.g. to rank them by their prestige or their influence. Subnetwork level methods look at groups of actors and decompose the network into subnetworks of cohesive actors or sets of actors with similar attribute properties. Finally, network level methods examine the network as a whole and provide results that describe the entire network.

On the actor level the CAT provides four centrality measures:

- Degree Centrality: measures the number of connections of a vertex
- Closeness Centrality: the inverse of the sum of the distances to all other nodes
- Inverse Distance Closeness Centrality: similar to closeness centrality but also defined for unconnected graphs
- Betweenness Centrality: measures the fraction of all (shortest) paths between all other vertices that pass a specific vertex

The subnetwork level analysis consists of network decomposition into (strongly) connected components and the detection of cohesive subgroups.

Finally, the CAT provides three network level measures

- Average degree: average number of connections of all vertices

- Density: fraction of all possible edges that do exist
- Diameter: the largest geodesic distance between any pair of vertices

The semantic layer of the CAT will provide the analysis semantics, i.e. an interpretation of the raw analysis results. In the emergency response scenario the community analysis tool provides social intelligence for the following use cases.

- In an emergency situation the emergency response authorities receives masses of – partly conflicting - information input from citizens. Some people exaggerate when describing the real seriousness of the situation. The emergency response staff need support in prioritizing input processing and dealing with conflicting information. The CAT analysis provides information about community relevance of WeKnowIt users. This information can be used to classify information submitted by this user.
- The automated image tagging methods for audio, image and video uploads to the WeKnowIt system request information about the uploader to decide in conflicting situations which tag to use. The CAT provides information about similar users to the current uploader so that the tagging methods can exploit the knowledge about uploaded media item of these similar users to find tags.
- WeKnowIt stores family networks to broadcast situation/location information of users in their family networks. WeKnowIt forwards emergency cases immediately if email addresses or phone numbers are available. Users can check their family member statuses and save time and don't put extra load on the stressed communication networks.
- In the recovery phase of an emergency citizens will deal with problems that they have in common with fellow citizens. WeKnowIt can connect users with other users in their (extended) friendship or neighbourhood network which have the same problem to facilitate private emergency recovery efficiency.

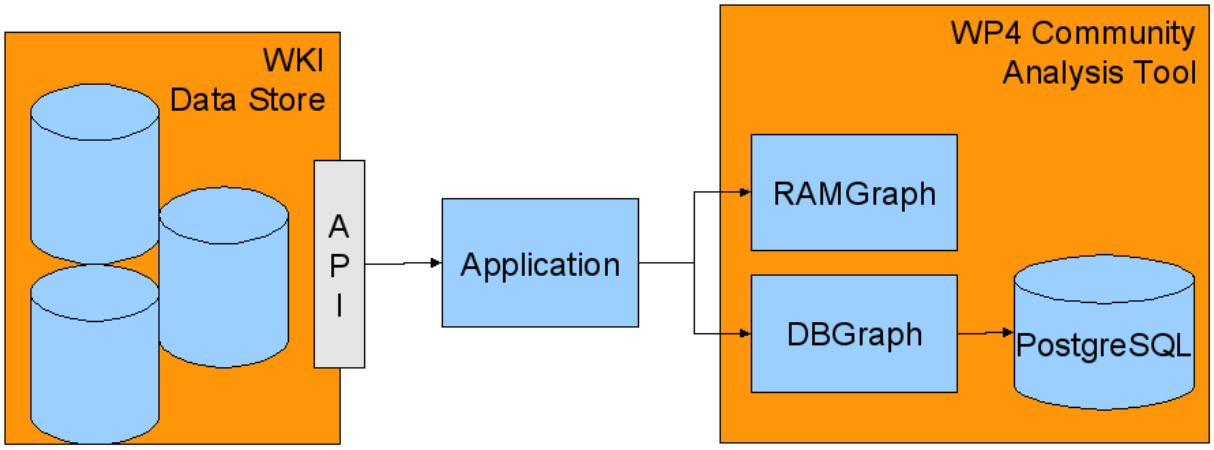


Figure 9 - Data flow towards graph data structures

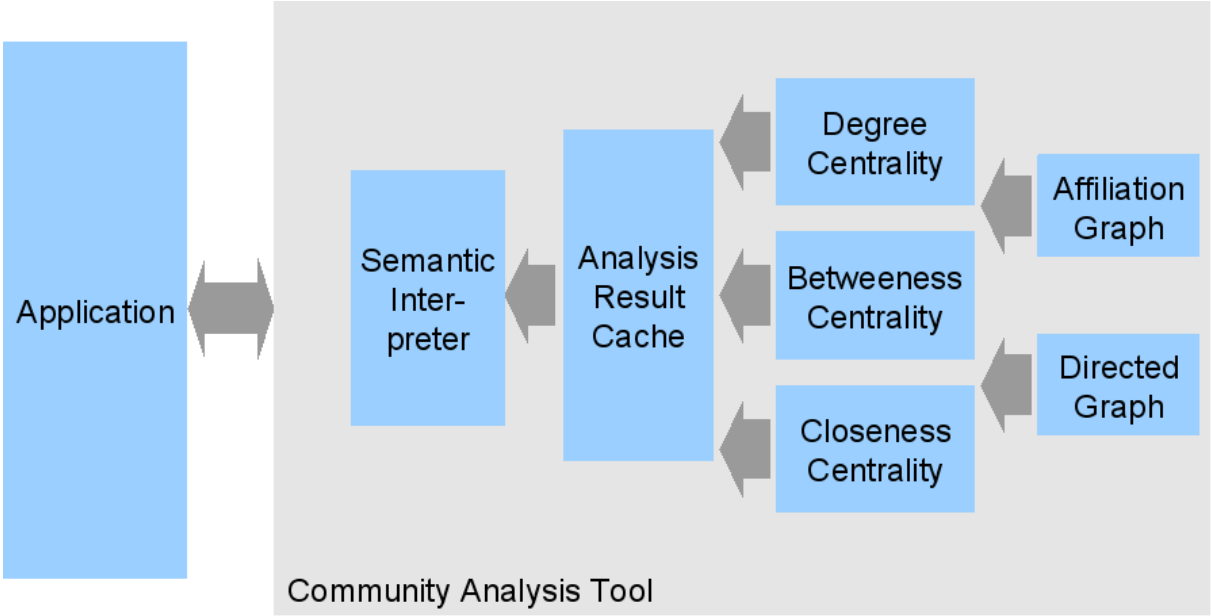


Figure 10 - Architecture of the CAT

5.5. Organisational Intelligence

Goal of organisational intelligence is to develop a knowledge management methodology and provide support for this methodology in the WeKnowIt project. The methodology shall allow for modelling and communicating collective knowledge gathered and processed from the different layers of intelligence, namely the personal intelligence, media intelligence, mass intelligence and social intelligence. This knowledge management methodology is generalized from specific perspectives of the use cases considered in WeKnowIt. It is also applicable to other use cases not considered in the project.

The basic idea is to model the collective knowledge in terms of events. The rationale behind this is that humans think in terms of events. Thus, events provide a natural abstraction of happenings in the real world. We are currently developing a common model of events, called the Event Model F[7]. The event model F is applicable to arbitrary domains, thus it is applicable also to the emergency response and consumer social group use cases of WeKnowIt. In this Event Model F [12], we define events as something that is said to "occur" or "happen". Consequently, events are perduring entities that unfold over time. The ontological competitors of events are objects. In contrast to events, objects are said to "exist". They are enduring entities that unfold over space. Events and objects require each other; events have participants, i.e. the objects, whereas objects are in time and by this bond to events.

The event model has to fulfil a set of different functional requirements (see [13]) that are serving different aspects of events, namely the constitutive, temporal, spatial, experiential, and structural aspects. Besides these aspects of events, a common event model shall also provide for different interpretations of events.

- Constitutive Aspect: The constitutive aspect describes the living and non-living objects participating in an event such as people, animals, and other material objects.
- Temporal Aspect: The temporal aspect covers the temporal extension of an event. It can be modelled using absolute or relative representations of time.
- Spatial Aspect: The spatial aspect is in charge of capturing the spatial dimension of objects participating in the event. This can be also modelled using absolute or relative positioning.
- Experiential Aspect: The experiential aspect comprises the annotation of events with sensor data such as media data.
- Structural Aspect: The structural aspect considers the arrangement of events in *mereological*, *causal*, and *correlative* relationships. Events are usually made up of other events. Thus, the common

[7] F = E + 1, a homage to event model E by Westermann and Jain

event model shall support the modelling of mereological relationships between events. Causality requires the modelling of causes and effects and should support the integration and use of different causal theories. Correlation refers to two events that have a common cause. While causality is very difficult to discover and, hence, often unknown correlation is typically easy to observe.

- Event Interpretations: Structural relations between events such as causality and correlation can be a matter of subjectivity and interpretation. For example, in a law-suit the parties involved may each claim that the other one is at fault. A common event model should be prepared to support such different interpretations of the same event.

For the design of our event model F, we initially conducted an extensive analysis of existing event models and studied related work on events in philosophy, linguistics and cognitive sciences. We choose the foundational ontology DOLCE+ Ultra Light (DUL) as modeling basis for our event model as it follows a pattern-oriented approach for ontology design. DUL is based on the foundational ontology DOLCE. However, it provides more intuitive naming and concept definitions. It also already defines upper classes such as Events and Objects. The event model is fully specified and documented in a technical report (see Scherp et al., 2009). We are currently also developing an API that allows other work packages to directly use the event model. To evaluate our event model F, we plan to employ it among others in a knowledge management tool that is currently being developed.

This knowledge management tool is designed in a modular fashion such that it can be applied for both use cases of WeKowIt. Currently, we are focusing on developing a concrete instance of the tool for the emergency response use case (see Scherp, Schwagereit, Ireson 2009). The functionality that shall be provided here are (see Functional Requirements in Section 2.3):

- Management of organisational structures: Setting up and maintaining emergency response team.
- Task management: Allocating the work within the emergency response team and defining, describing and assigning tasks to persons.
- Incident log: Capture, represent and share the current state of the incident in terms of events.

A mock-up of the knowledge management tool interface is presented in Section 7.5.3.

6. User Rights System

6.1. Non-registered user

Non –registered users will be able to view the available information but will not be able to make use of advanced functionalities like

- Submitting new content
- Tagging or annotating information

6.2. Registered user

When registering users, they will be assigned to a pre-defined user group:

- Personal users, i.e. normal citizens
- Organisational users, i.e. ER personnel of different qualifications and experience

While all personal users will have the same rights, the situation changes for Organisational users, as in order to guarantee information control, at least two different user roles must be taken into account:

- An Admin ER user, that has the moderator rights to see all the information available, to use all the tools, to approve or reject information for the public, to send information to the public
- A standard ER user.

7. User Frontend

In the following sections we will present the design work that has been conducted by all the partners for the first prototype of the WeKnowIt application for ER use case.

We will first of all explain the design principles that guided the work, followed by example mock-ups that refer to specific parts of the scenario defined in D7.1 or to the use case in general.

The design process will continue following the iterative design methodology: sessions will be set up with users to walk through the mock-ups and provide feedback. The feedback will be incorporated in new mock-ups. It is envisaged that at least 2 iterations will happen before finalising the design of the first prototype.

7.1. Design

The overall application has been designed to meet the user requirements outlined in Section 2, with a particular attention to the High priority User Requirements for both user groups.

Moreover, all the user requirements for the organisational users must be met, with a particular attention to manage efficiently the user rights and visualisation, so to offer the right content to the right user at the right time.

Following the scenario presented in D7.1, a diagram has been derived to describe the flow of the application, identifying the main functionalities/services corresponding to each step of the scenario and matching the requirements.

In the following diagram (Figure 11) all the tasks are based on the concept of event and users. Events are information and feeds containers. They can host messages, pictures and so on.

The tasks in the diagram are colour-coded to express their specificity and relation with respect to the WeKnowIt application.

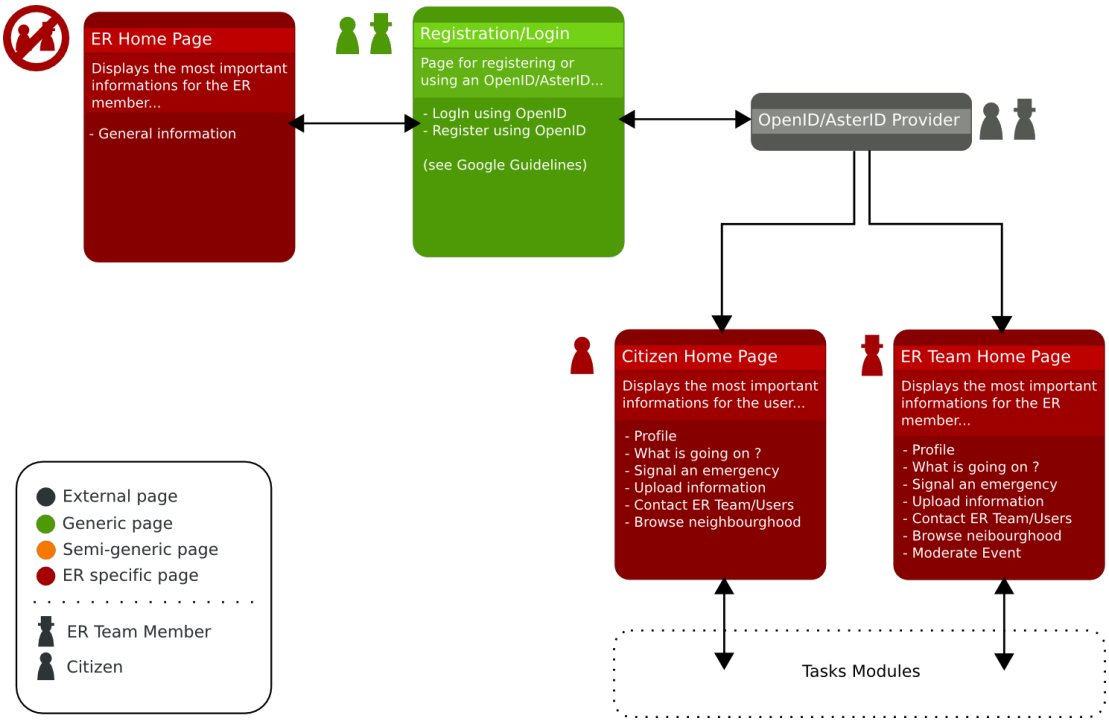


Figure 11- Diagram of Pages Flow: Login and Profile Pages

Messages, users, events (content) are all hosted in a generic container or component with different attributes. These containers can be linked together in an oriented graph structure (parents/children). The previously mentioned Sparks framework will provide this data structure.

Each of the component content is defined as an RDF statement. The content restricts and defines the relations between components. For instance it may not be possible to link sound with picture. Linking a media with a user create an ownership relation. Linking a user to an event can be interpreted as a feed subscription. The link description is also done using RDF.

As a consequence, each piece of information can be linked to anything and it is easy to provide generic, flexible and extensible UI components.

The following diagram describes how Personal Intelligence and Organisational Intelligence tasks and components are interrelated in the design process.

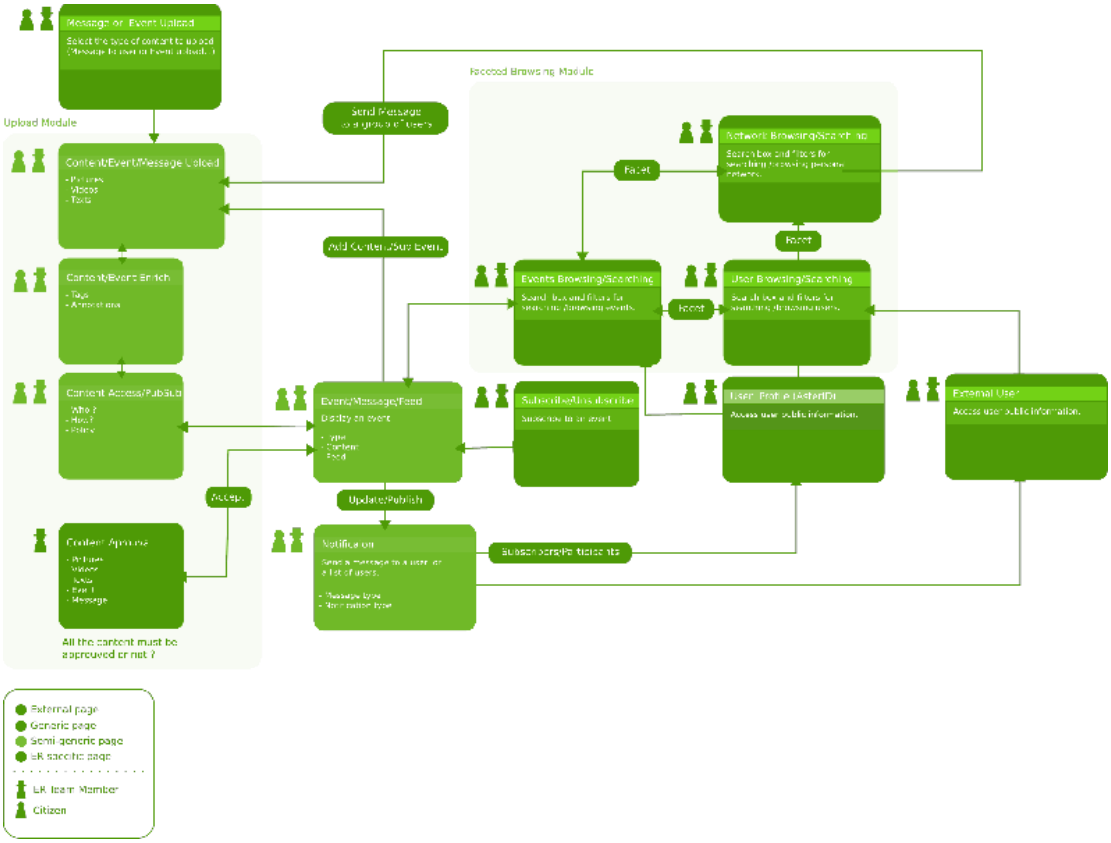


Figure 12 - Diagram of Pages Flow: Tasks/Components

The page flow diagram highlights the key role of the event and user model. Each user is linked to an event as a participant. Events cannot exist without users. The event model is used in this context for mapping the actions of the users rather than the Emergency events. However, these two types of events are related through the uploaded content and the participants.

7.2. Login

WeKnowIt will adopt a login system based on AsterID, an extension of the OpenID protocol. The login interface will provide the user with an easy way to register their profile to the system and access the information.

This feature is also used to control the information available to each user role.

7.2.1. Summary of Main Features

For the UI implementation, the idea is to use a specific approach based passgraph instead of passwords. This approach, called GraphPad [14], is designed for easing the user authentication process. This is particularly relevant for the ER user case where .

Moreover, pictures have others advantages over usual passwords:

- They are easier to remember;

- Passgraphs are usually less weak compared to passwords (usual names);
- They cannot be attacked using dictionary attacks.

Some disadvantages exist compared to passwords [15] but they are minor compared to the GraphPad approach. Users tend to choose smaller passgraph compared to password, but having a minimum length passgraph easily solves it. The probably biggest problem of using pictures instead of characters is concerning the cost of downloading pictures on a mobile phone sin it may take a while before being able to use the page.

For securing the login process, it is also possible to use session dependent "picture codes" with limited lifetime so the passgraphs are never exchanged in plain text.

The use of passgraph is particularly suitable for mobile phones and touchscreen where the keyboard is limited. The idea is to display the symbols using the same layout of the phone keyboard. By doing that, it is easy and fast to type a passgraph. For each round, there are 144 possible symbols (more than the symbols of a computer keyboard).

7.2.2. Example: User registration from a mobile device

The following scenario is considered.

Mark, a Sheffield citizen, is setting up a personal account on WeKnowIt

According to the first page flow diagram, the user registration is done using an OpenID form displayed on the home page of the ER webpage. Because WeKnowIt require some extensions to the OpenID protocol, WeKnowIt will provide an OpenID front-end.

There are two different cases for user registration. When Mark wants to login in WeKnowIt for the first time, he may use an external AsterID (the OpenID extension used by WeKnowIt) account or create a new one on the WeKnowIt server. Since the WeKnowIt AsterID implementation will be the only one existing at the moment, an AsterID user profile will be automatically created when Mark will want to register a new account. An interface will be provided to Mark for writing his personal information, such as his name and email.

Description

The User interface for filling the personal information of the user should be similar to the OpenID profile interfaces used by OpenID providers. However some level of interface integration could be provided since

WeKnowit is an AsterID provider and a relying party at the same time and, it is expected at the beginning that all the registered users accounts will be hosted in the WeKnowIt AsterID server. In future, a simple interface could tell the user that he can register an AsterID on another account.

Screendesign

Figure 13 - User registration on a mobile device

7.2.3. Example: User login on a mobile device

When the login has been already set up, the user just has to type their passprhase or use OpenID to log in.

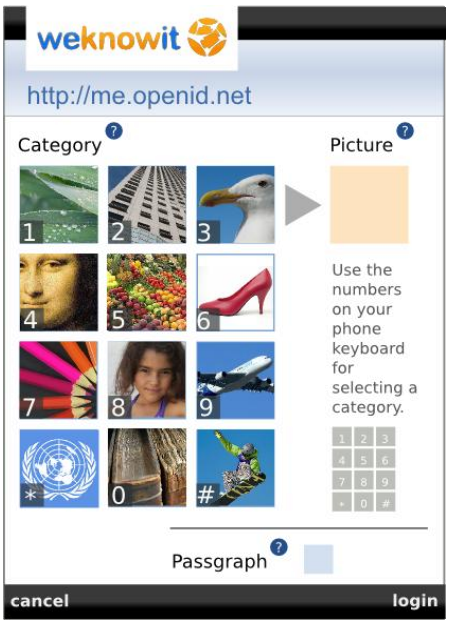
Description

When a user wants to login in the system, he is required to give his OpenID login.

The interface for setting up a user login is shown in Figure. Under each number a picture should be displayed. An example of category could be animals, and the symbols of the category different animals such as a cat, a spider... For improving the security against phishing a seal could be used [16] (as the ones provided by Yahoo!).

For avoiding the cost of downloading pictures, the interface can only consider "gestures" instead of pictures. For instance, the categories becomes colours and the symbols become patterns (coordinates: red(b,1)). It will be easy for people to remember their code because of its spatial pattern (than can not be applied for a standard password).

Screen design



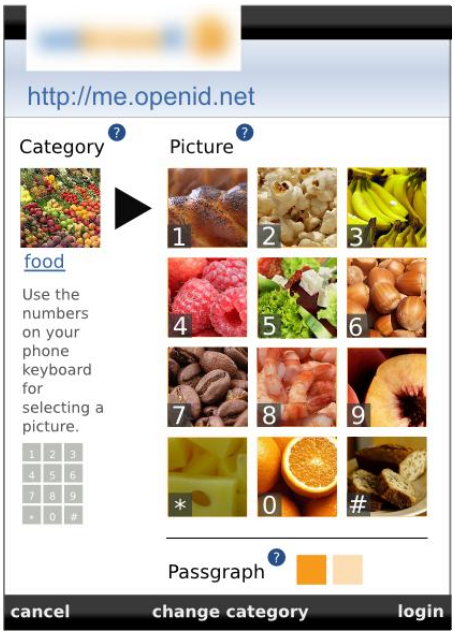


Figure 14 - User login from a mobile device using passgraphs

7.3. Homepage

The homepage has been designed as a first point of contact for every type of users, offering a summary of the available content and of the installed widget/applications.

7.3.1. Example: Homepage for ER user

Accordingly to the scenario in D7.2 an admin ER team member needs to have a dedicated interface that shows all the available information and allowed to manage all the available knowledge, by for example viewing the status of the other personnel, switching between applications/functionalities, viewing a summary of the event.

Description

As shown in Figure 15, the interface is divided into two main areas, a top area with a tool bar that allows the user to access all their personal information (Profile, Network, Inbox), a application taskbar, illustrating all the available applications/functionalities and a quick summary of the status. In the bottom area a summary of the event and the available content is displayed, with the possibility to filter the visualised content using any of the facets.

Screen design

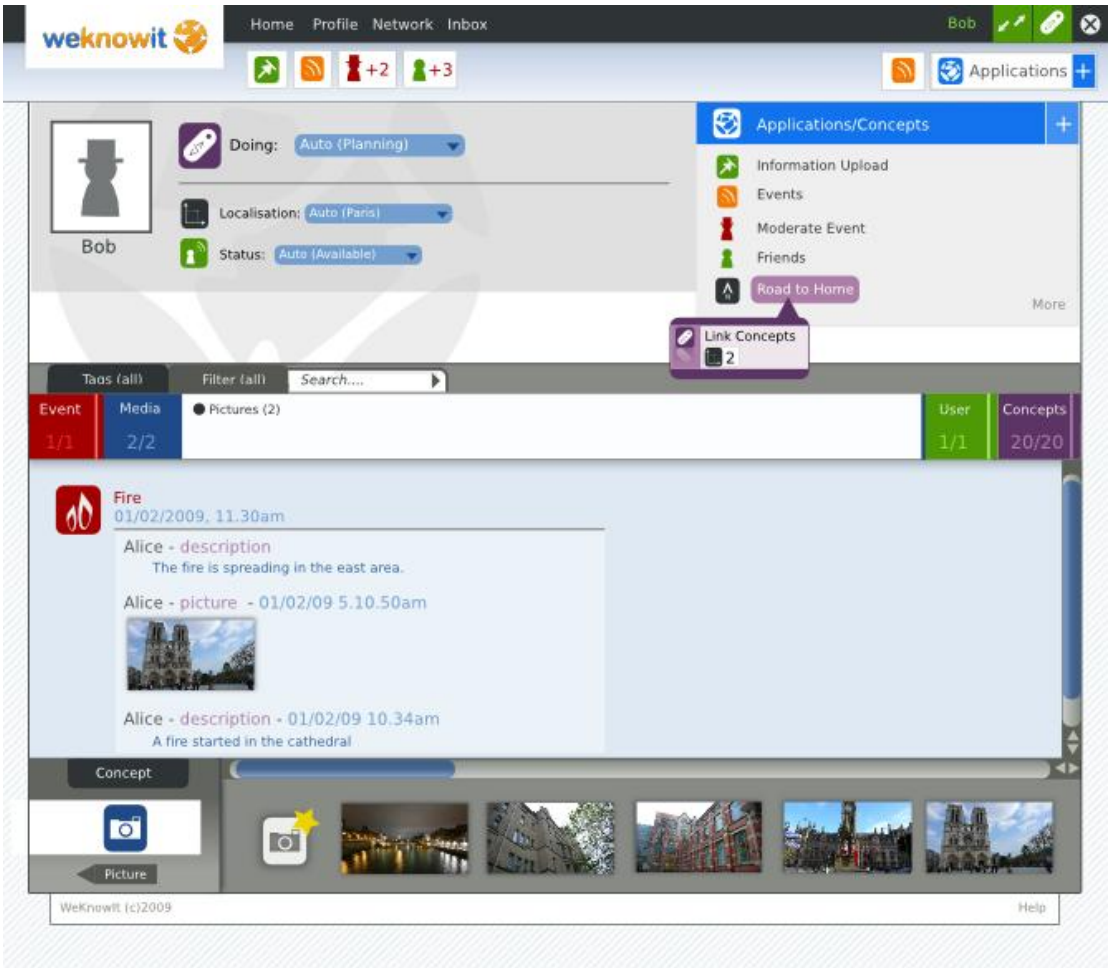


Figure 15 - Homepage for ER user

7.4. Ontology-based annotation

A very simple and intuitive way to gather ontology-based annotations from the end users is to adopt a form-based interface, asking the user to answer simple questions related to the content or the event.

As explained in Section 7.4 the forms are easily customisable, thus giving maximum flexibility and expressivity to an organisational user.

7.4.1. Summary of Main Features

The form-based visualisation will be embedded in the user interface. When the user uploads an item of content (image, video, text) a link will be displayed inviting him to provide additional metadata. For a citizen this step will be voluntary, as someone involved in an emergency may not want to take the time to answer to form questions.

The form-filling step may instead be mandatory for an organisational user.

7.4.2. Example: Form-filling for an organisational user

Andrea hears the mobile phone text message and reads it. Immediately she starts taking pictures and uploads them to E-WeKnowIt.

Description

A from designed by an admin ER member is displayed when Andrea uploads her pictures on the WeKnowIt server. The form is very simple and offers the user a guided way to choose the most correct answer (thus ensuring the content is annotated very precisely).

When a precise, pre-defined answer is not possible the user is offered the possibility to enter free text.

Screen Design

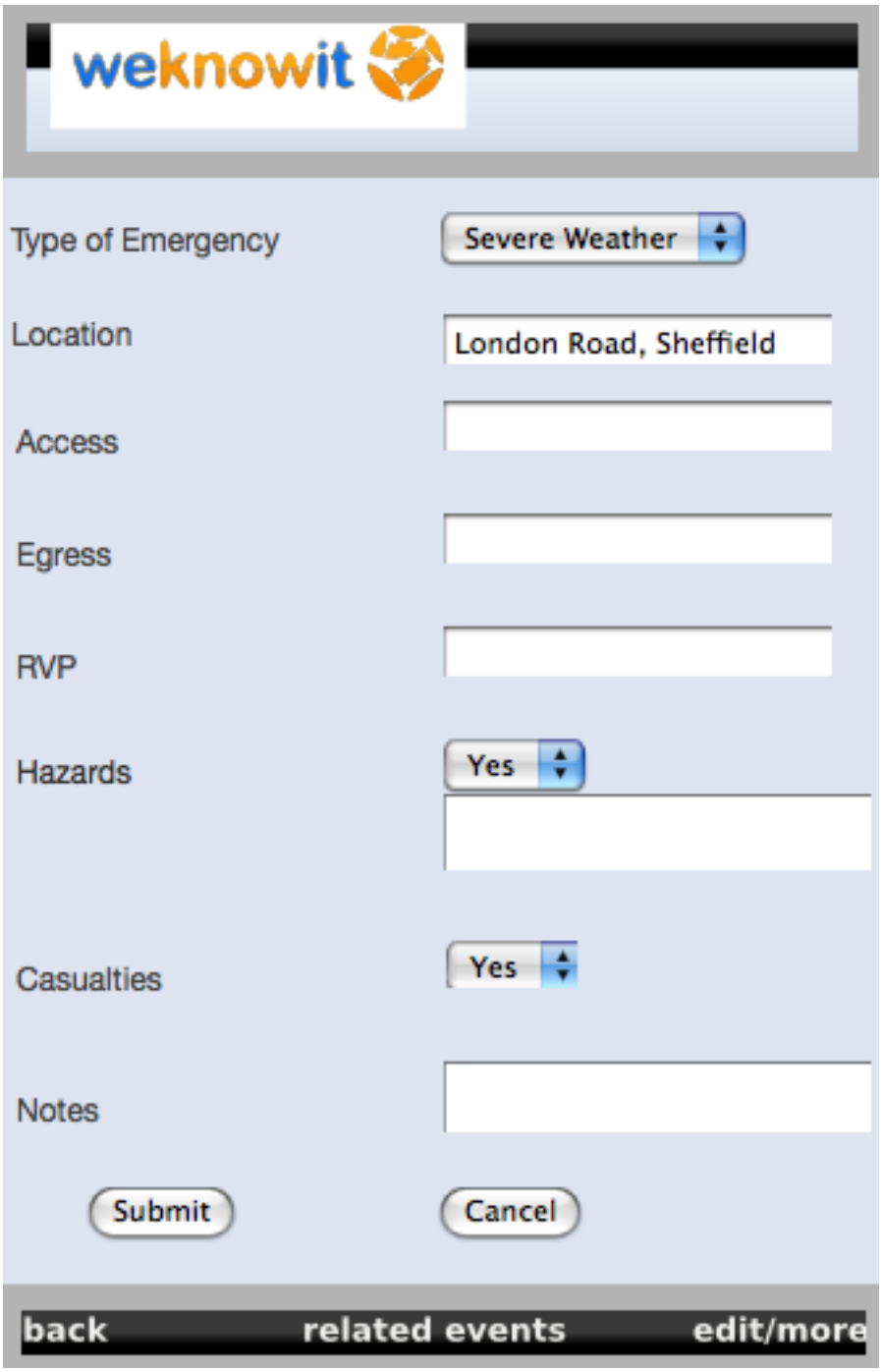


Figure 16 - Form providing ontology-based annotations in background (invisible to the user)

7.5. Browse

WeKnowIt will use faceted browsing techniques to offer the user an easy way to navigate the information space.

7.5.1. Summary of Main Features

The faceted visualisation will be initially focusing on the geographical and temporal dimension, with the aim of extending the facets after the first evaluation phase, if deemed useful by the users.

While designing the browsing interface the idea was to consider an event as a flow of information and data, thus offering the user the possibility to play a movie of the event, pause, zoom in and out, change the granularity level and so on.

7.5.2. Example: Event-based faceted browsing on a mobile device

The following scenario from D7.2 is considered.

In the ER Silver Team John has received a text message that notifies that new information has been added in E-WeKnowIt. This information is the aggregated data of the different inputs provided to E-WeKnowIt by John's colleagues, silver and bronze members, as well as data provided by citizens.
When opening the main page John chooses to view the new information using a geographical visualisation. John can see the location of the photos. This helps John in having a better overview of the situation.

Description

The map interface is divided in two sections: the first one represents a map of an area with localized events represented by standardized symbols. The second part represents the timeline of the events. The event timeline is divided in two sections. On the top, the user can select a day or "play" the events. The second option enables the user to view the evolution of a situation: the symbol of the map move and change. The second part displays the events of a day with the beginning and end date of each. When a user selects an event, its description is opened.

The event view shows a simple description of an event with the timeline. Two possibilities could be considered: the first one displays the timeline of all the events whereas the second one shows only the timeline of the current event and enable the user to navigate easily in each item.

The navigation through the interface uses the keyboard of the phone (the numbers) or touchscreen interaction for accessing the map. The buttons *,0,# switch the behaviour of the numbered keys: zoom, selection of an event, navigation in the map. The numbers could also be used in multimodal interface with speech input.

Screendesign

Figure 17 - Map-based visualisation of an event content on a mobile device

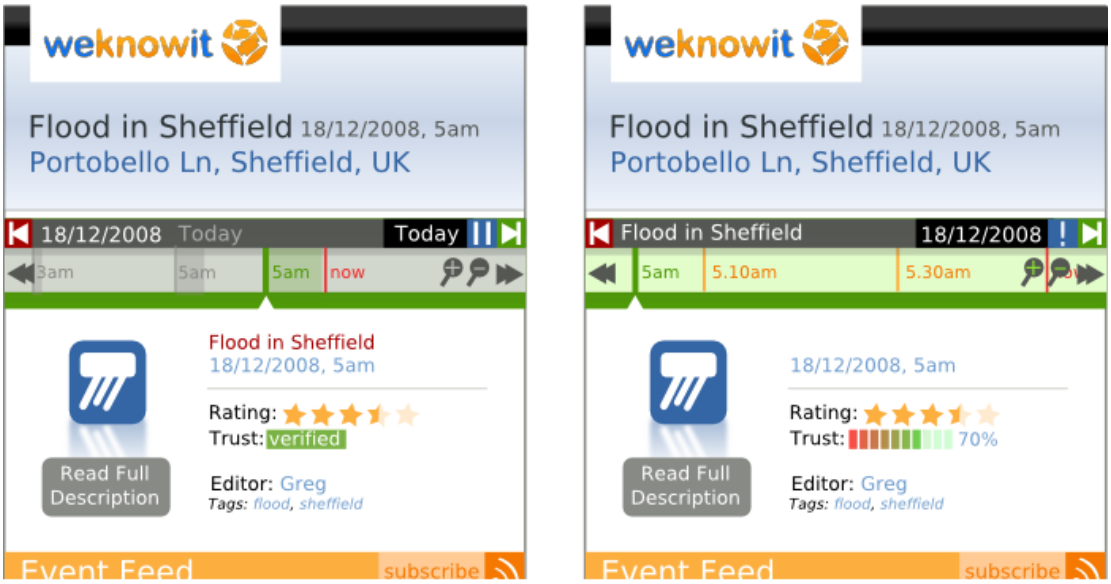


Figure 18 - Event summary on a mobile device

7.5.3. Example: Event-based faceted browsing for an ER admin user on a desktop computer

The following scenario from D7.2 is considered.

> When opening the main page John chooses to view the new information using a geographical visualisation.
>
> John can see the position of the persons that sent the information and the location of the photos.
>
> This helps John in having a better overview of the situation.

Description

As the user is accessing the system from a desktop interface, more functionalities and visualisations are displayed.

In this view an admin organisational user is presented a summary of the event visualised accordingly to the temporal and the geographical dimension. The event is presented as a timeline, that the user can play, pause, rewind and forward. While the timeline plays the user can visualise the change of information in time; the user can also zoom in and out, thus changing the granularity to which the information is displayed.

Different facets are available to refine the amount of information displayed at a given time, thus leaving it up to the individual preferences how much information is visualised at once.

The user has also the possibility of visualising an item of content in detail: as this interface is thought for an admin ER user, they have the possibility to edit the tags, remove them, change the priority, publish the information for all the other users to see.

Screendesign

7.6. Task Management

7.6.1. Example: Task Management on a desktop computer

Description

The screenshot in Figure 19 depicts a view on the task management module of the WeKnowIt knowledge management tool for emergency response. The head of the emergency control centre Sarah Armstrong is logged in. In her task view, she sees on the left hand side all tasks she is working on. On the right hand side, an overview of tasks issued by Sarah is shown that she delegated to her colleagues. The details of the task "Water level at Downing Street" are shown. They include a task description and the time and date the task was issued, a task priority and

the executor of the task. Some water levels are entered here as task result (indicated by the filled star) that Sarah can constantly check.

As said above, the knowledge management tool is designed to be applicable in the consumer social group use case as well. Here, the functionality described above can be reused in the following manner:

- Management of organisational structures: Defining the group of tourists/friends that travel together.
- Task management: Conduct the typical activities for trip planning etc. Supported by checklists and guidelines that define standard-tasks to be conducted.
- Incident log: Trip diary in terms of events.

ScreenDesign

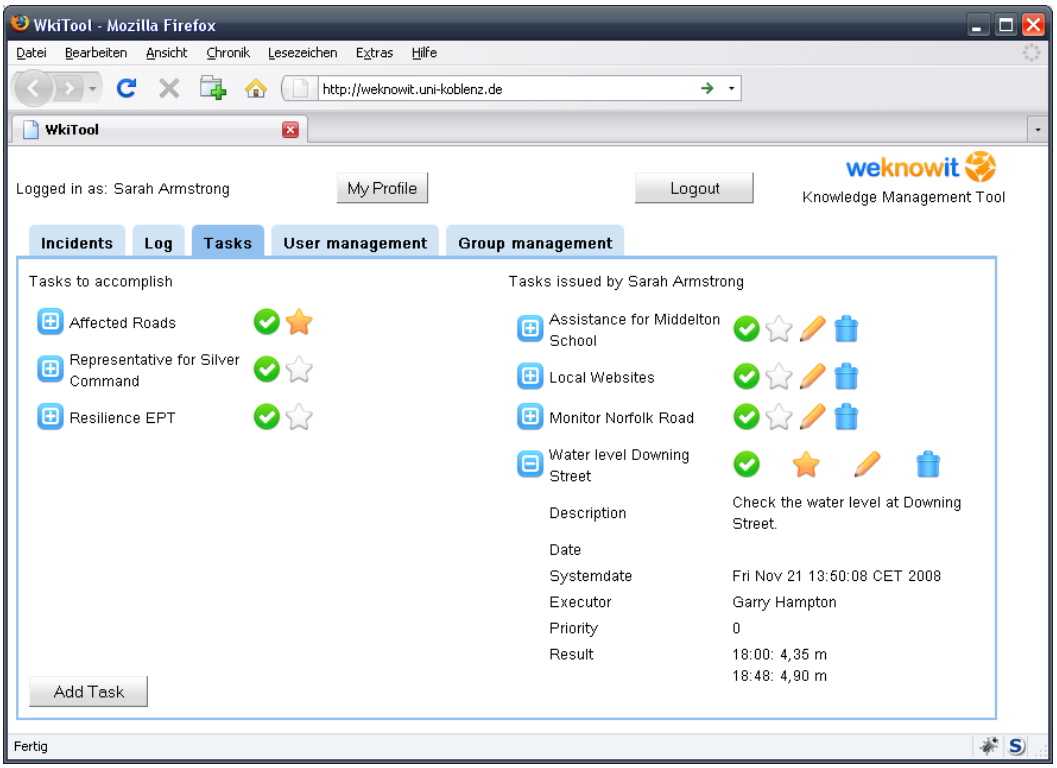


Figure 19 - Screenshot of Task Management View in the Knowledge Management Tool for the Emergency Response Use Case

8. User Participation

ER cases are of special nature compared to – for instance – consumer use cases. In an emergency event, the people that can contribute to the raising of knowledge about the incident at hand are limited to the scope of those citizens who are directly impacted (e.g. being in the situation of dealing with the event), or have a less intimate connection (e.g. happening to be an onlooker to an incident, a communicated relative of a person in emergency need, etc.). This scope of potential contributors to WeKnowIt knowledge base may become even more restrained, taking into account those people's willingness to participate, the exact situation as permitting the use of an accessing terminal, among many other factors. For instance, actually affected users might never bother to use WeKnowIt, as their main and foremost concern would be to escape from the event in which they got entangled.

A rough classification of users can be made based on the combination of some generic attributes shown in Table 3. Any combinations of these attributes dictate the "Level of Participation Level or Response" that users may provide.

Impact Level	**High:** Directly affected
	Low: Affected not in terms of e.g. life danger, but e.g. delays due to increased traffic
	None. No impact at all
Impact Type	**Direct:** The person of interest is actually affected.
	Indirect: People related to him affected e.g. relatives
Location	**In the area** of the event
	In the vicinity of the area of the event
	Outside the area of the event
Time	**At the time** of the event
	After the event occurrence
	Before the event if warning notifications have taken place e.g. bomb explosion
Event Knowledge	**High:** Has a high degree of knowledge of the event or its side effects
	Medium: Medium knowledge.
	None: No knowledge about the occurrence/Side effects of the event
Willingness to learn	**Absolutely** needs/wants to know more
	Medium level: May require some info
	Not willing to learn more

Participation Level	**High:** Willing to contribute irrelevant of e.g. any incentives
	Medium: May contribute under certain circumstances e.g. incentives
	Low: Would never contribute

Table 3 – Factors dictating the level of citizen participation to WeKnowIt, in an ER case

The collection of as much information that describes the situation of an emergency event is a necessary requirement for organising and planning an ER operation, as well is for the successful use of the WeKnowIt ER services. Therefore, taking into account the generally narrow scope of the people involved with an emergency situation, it is of definitive importance that the WeKnowIt application receives as high usage penetration as possible. This could be accomplished through raising the level of participation of as many involved people as possible. There can be several ways or mechanisms that can influence the participation level of the rightly targeted citizens. For instance:

- Notification of the actual event
- Notification/advertising of WeKnowIt existence
- Free subscription / Instant accessing by people unregistered to WeKnowIt, but who may be in the area of the emergency event
- "Push" of information useful or necessary for the safe evasion (e.g. traffic information, evacuation directions, etc.)

In such a way, for instance, a user with a "participant's profile" of

- Impact Level: medium
- Impact Type: direct
- Location: in the vicinity
- Time: now
- Event Knowledge: none
- Willingness to learn: medium

could – through stimulus mechanisms – change into

- Impact Level: medium
- Impact Type: direct
- Location: in the vicinity
- Time: later
- Event Knowledge: high
- Willingness to learn: high

thus effectively be triggered to participate in the WeKnowIt community.

Therefore, it is important to find mechanisms to identify the right target group (on a per case basis) expected to form the WeKnowIt community.

8.1. Stimulus mechanisms for ER participants

ER events are of special nature (random location, time, type of users) and as such a number of concerns should be considered:

- It is highly probable that citizens present in the vicinity of the emergency event are not WeKnowIt users. (WeKnowIt applications expected to mature after the next 4 years.)
- It is highly probable that people present are not IT/Telecom–literate, thus reduce the number of WeKnowIt potential users.
- Most people are highly probable to contact relatives or friends as a first priority.
- Those that contact ER services and are notified on the WeKnowIt, may still choose not to use it, but would rather escape from the event.
- To fully utilise WeKnowIt resources, a critical mass of users must exist

All the abovementioned issues point to the fact that the level of response from people may be really limited. In order to make more probable that, in emergency cases, WeKnowIt will actually be used by those people that can share invaluable information about the event, certain targeting strategies and incentives should be committed. This should also be seen in conjunction with the type of users, as described in the previous section. There are a number of issues that WeKnowIt owners, possibly in collaboration with mobile network operators, should consider to exploit the full capabilities of WeKnowIt, as for example:

- **Advertise**: When emergency occurs, advertise the existence of the application, its characteristics and its use, along with the information that an emergency has occurred.
- **Raise awareness**: Even if customers know that such an application exists, provide incentives e.g. "if you participate you will have access to provided content e.g. traffic info".
- **Subscription-free**: Due to the nature of these events, a user will expect not to pay for its use. Thus, offer its use for free.

- **Level of awareness**: Inform a large number of users, so even with a generally low response rate, we manage to create a critical mass of users. A solution can be to broadcast /multicast the information to the vicinity of the area.
- **Network resources**: In these types of events, (radio) network resources are highly clogged up, so an optimised utilization of resources should be pursued; for instance, a Multimedia Broadcast/Multicast Service (MBMS) type of alert could be used.
- **User Targeting**: Dynamic/Smart user selection can occur through profile pre/post-monitoring. Pre-monitoring is a challenge, but for post monitoring, user profiles/attributes (e.g. user's location, terminal type) could be monitored in order to best target and notify the most suited citizens for prompting them to contribute with information relating to a specific event.
- **Timing**: Timing is also crucial in relation to the type of users and the information required. A fast and efficient means of user alert and response would increase the exploitation of WeKnowIt capabilities.

9. Conclusions

There are many different types of emergencies; ones brought about by forces of nature such as avalanches, floods, droughts, earthquakes or man-made emergencies, e.g. train and plane crashes, pollution and terrorist attacks. These emergencies can vary in terms of scale, both in severity and affected location. In small scale emergencies, only a few people and organisations may be involved, typically only local authorities such as the city council, police and fire department. During large-scale emergencies several hundred organisations can be involved, and thousands or even millions of individuals; as was the situation after the Tsunami in the Indian Ocean 2004.

The Emergency Response (ER) team's decision-making process primarily involves the allocation and coordination of resources, but also maintaining effective communication between the agencies involved, the decision/command chain and the affected individuals. During an incident, the ER team receives information from multifarious sources (e.g. the emergency services, other local authority bodies, government bodies, broadcast services, affected individuals, and others). The management of the mass of information is crucial in aiding the decision-making, ensuring, as far as possible, that the responders have full situational awareness to make informed decisions.

In order to better support the ER activity, a specific use case has been investigated as part of the WeKnowIt Project, to demonstrate the wide applicability of WeKnowIt methodologies and technologies.

Currently ER focused projects and applications are mainly based on personal and organisational intelligence, supporting only visualisation of the available information without any intelligent analysis and are targeted either at the Emergency Response personnel or at citizens; WeKnowIt instead aims to exploit the power of Collective Intelligence to benefit both citizens and ER personnel, thus making available huge amount of real time information and more sophisticated intelligent analysis.

In this deliverable revised user requirements, with an in-depth analysis of the implied functionalities and services and a detailed description of the envisaged application were presented.

In particular the requirement analysis focused on outlining the differing and matching requirements between Organisational and Perosnal Intelligence and their relative priority, as guidance for the design process.

The envisaged first prototype was described in details, with particular attention to the application architecture, the back-end systems (contributions that each workpackage (each intelligence layer) provides and how this meets the user requirements) and the front-end user interface.

In order to ensure the integration between the Intelligence Layers, each workpackage has focused on some parts of the user scenario identified in D7.1, trying to explain how their methodologies and technologies can help. Moreover some cross-workpackage services and methodologies have been investigated, as for example a common event-based model to describe semantically the objects at the abstract level of content or at the more practical level of UI. Also an emergency Notifications ervice has been devised, that combines services provided by WP1, WP4 and WP5 to ensure quick delivery of status notification messages during an emergency to members of a family, team or organisation.

Although this deliverable mainly concerned the work done for the Emergency Response use case, the methodologies and technologies adopted and the design process and results can easily be extended to other use cases, in particular to the Consumer Social Group case study, as they have been designed and developed in generic terms, thus making them easily customisable for specific domains.

10. References

[1] Kim, S., Jang, Y., Mellema, A., Ebert, D.S. and Collins, T. (2007) Visual analytics on mobile devices for emergency response. VAST '07: Proceedings of IEEE Symposium on Visual Analytics Science and Technology (Sacramento, CA, USA), IEEE Computer Society: Los Alamitos, CA, USA, 2007; 35-42.

[2] Schneiderman, B. and Preece, J. (2007) 911.gov, in Science 2007, vol.315 page 944

[3] Berrouard, D., Cziner, K. and Boukalov. A. (2006) Emergency Scenario User Perspective in Public Safety Communications Systems. Third Information Systems for Crisis Response and Management Conference (ISCRAM2006), Newark, New Jersey, USA, 14-17 May 2006.

[4] de Leoni, M., Mecella, M., De Rosa, F., Marrella, A., Poggi, A., Krek, A. and Manti, F. (2007) Emergency Management: from User Requirements to a Flexible P2P Architecture. In Proc. of the Session on Flexible Service and Data Management Platforms for Crisis Response at the 4th International Conference on Information Systems for Crisis Response and Management (ISCRAM), Delft, the Netherlands, May 13th-16th, 2007

[5] Robillard, J. and Sambrook, R.C. (2008) USAF Emergency and Incident Management Systems: A Systematic Analysis of Functional Requirements. http://www.uccs.edu/~rsambroo//Research/EIM_REQS.pdf

[6] Diehl, S., Neuvel, J.M.M., Zlatanova, S. and Scholten, H.J. (2006) Investigation of user requirements in the emergency response sector: the Dutch case. Leerstoelgroep Landgebruiksplanning, 2006

[7] D7.1

[8] Gregoire Burel. Toward User Ubiquity --- Semantic Agent Architecture for Federated Identities, ESWC09 (PhD Symposium)

[9] Ravish Bhagdev, Ajay Chakravarthy, Sam Chapman, Fabio Ciravegna, Vita Lanfranchi: Creating and Using Organisational Semantic Webs in Large Networked Organisations. In Proceedings of the 7th International Semantic Web Conference, Karlsruhe, Germany, October 2008.

[10] Steinbuch, Pitter A. 2001. Organisation. Friedrich Kiehl Verlag GmbH. Edition 12.

[11] Chapin, Peter C. and Skalka Christian and Wang, X. Sean. 2008. Authorization in Trust Management: Features and Foundations. ACM Computing Surveys, Vol. 40, No. 3, Article 9.

[12] A. Scherp, T. Franz, C. Saathoff, S. Staab, F---A Model of Events based on a Foundational Ontology, University of Koblenz-Landau, Germany, January, 2009.

[13] A. Scherp, F. Schwagereit, N. Ireson, Web 2.0 and Traditional Knowledge Management Processes, Workshop on Knowledge Services & Mashups, at 5th Conference Professional Knowledge Management, Solothurn, Switzerland, March, 2009.

[14] Gregoire Burel, Elizabeth Cano, Vitaveska Lanfranchi, Fabio Ciravegna. GraphPad – User Friendly and Secure Login Interface for Mobile Devices Using OpenID , MobileHCI09.

[15] Xiaoyuan Suo , Ying Zhu , G. Scott. Owen, Graphical Passwords: A Survey, Proceedings of the 21st Annual Computer Security Applications Conference, p.463-472, December 05-09, 2005

[16] Naveen Agarwal, Scott Renfro, Arturo Bejar. Yahoo!'s Sign-in Seal and current anti-phishing solutions