SAGA: an Integrated Architecture for the Management of Advanced Emergency Plans

José H. Canós
ISSI-DSIC
Universitat Politècnica de València
jhcanos@dsic.upv.es

Abel Gómez
ISSI-DSIC
Universitat Politècnica de València
agomez@dsic.upv.es

Mª Carmen Penadés
ISSI-DSIC
Universitat Politècnica de València
mpenades@dsic.upv.es

Marcos R.S. Borges
Programa de Pós Graduação em Informática
Departamento de Ciência da Computação
Instituto de Matemática
Universidade Federal do Rio De Janeiro
mborges@nce.ufrj.br

ABSTRACT

Despite the significant advances that software and hardware technologies have brought to the emergency management field, some islands remain where innovation has had little impact. Among them, emergency plan management is of particular relevance due to their key role in the direction of teams during responses. Aspects like coordination, collaboration, and others are spread in plain text sentences, impeding automatic tool support to improve team performance. Moreover, administrative management of plans becomes a mere document management activity. In this paper, we present SAGA, an architecture that supports the full lifecycle of advanced emergency plan management. By advanced we mean plans that include new types of interaction such as hypermedia and advanced process definition languages to provide precise specification of response procedures. SAGA provides all the actors involved in plan management a number of tools supporting all the stages of the plan lifecycle, from its creation to its use in training drills or actual responses. It is intended to be instantiated in systems promoted by civil defense agencies, providing administrative support to plan management; additionally, editing tools for plan designers and tools for analysis and improvement of such plans by organizations are provided. Plan enactment facilities in emergency response are also integrated. To our knowledge, it is the very first proposal that covers all the aspects of plan management.

Keywords

Emergency Management, Emergency Plans, Architecture, Information System.

INTRODUCTION AND MOTIVATION

Research and development in emergency management systems have grown significantly in the last decade. On one hand, advances in hardware, especially the advent of a myriad of smart mobile devices, have brought solutions that allow the delivery of information to and from response teams from and to command control rooms in a way faster and richer than previous systems (Catarci, Leoni, Marrella, Mecella, Russo, Bortenschlager and Steinmann, 2011). Today, multimedia information is captured, transmitted and processed in a way that enhances situation awareness significantly. Sensor networks capture huge amounts of data that are crucial to improve early alerting systems to minimize damages (Galton and Worboys, 2011). And ambient intelligence systems combine such sensors with everyday objects to generate smart environments where multidirectional information flows are created.

On the other hand, software technology has evolved towards more intelligent systems able to process large amounts of data coming from the sensor networks, retrieve information from large collections of documents, and, especially, setting up the field of social applications that have demonstrated to be a new interaction paradigm with special influence in the organization of volunteering during emergency responses (White, 2011). Also, new media have brought

access to information into a new dimension where multimedia navigation spaces permit rich user experiences in an intuitive way.

In such a technology-enhanced universe, some islands remain in emergency management where innovation is still to arrive. Among them, emergency plan development and management is of particular relevance since the emergency plans are the essential building block of every response. Despite the sophistication of multimedia, and the expressive power of process languages, the operational aspects of responses are specified as natural language descriptions stored in traditional document-like files. It is time to apply new technologies to move toward a new generation of emergency plans that will be easier to develop, audit, maintain, and reuse, as well as more powerful in their support to responses. Facilities like personalized views for different response teams and civilians affected by the emergency, bi-directional multimedia information retrieval and transmission, workflow-like coordination, and others are considered key in future emergency response environments (Canós, Alonso and Jaén, 2004).

Making advanced emergency plans viable requires the development of management tools for editing and enacting these plans, among other tasks. In this paper we define the requirements and architecture of SAGA, a framework for the development and management of emergency plans. The SAGA acronym comes from the Spanish name corresponding to Self-protection Management Support System. Self-protection is the name that the Spanish Government has coined to name all the activities related to emergency management in public service organizations (NBA, 2007). SAGA supports all the actors involved in emergency plan management. First, it supports the administrative processes run at the Civil Defense agencies; examples of such processes are the registration of plans in a Plan Registry, and the definition of drills to check the correctness of emergency plans. Second, SAGA provides tools for editing emergency plans following a reuse-based approach and unifying structure and representation of these plans according to legal regulations. An third, the unification of emergency plan formats, along with the use of formal process languages to specify the response procedures permit the automatic enactment of plans in cases of actual emergencies or simulations.

This paper is structured as follows. We start with an enumeration of the requirements of a platform supporting integral plan management; most of them are derived from law, as we describe for the case of Spain. Next, we outline the overall architecture of SAGA, and describe in detail its three main components: the Self-Protection Digital Library, the Plan Generation System, and the Plan Execution Engine. Next, we provide some detail about the implementation of the SAGA architecture we are involved in, and a short discussion about their evaluation. Finally, we provide some clues about the intended future work on the SAGA development and improvement.

REQUIREMENTS FOR A SOLUTION

The Spanish Law on Self-Protection (NBA, 2007) prescribes the basic guidelines for emergency plan development and management. A summary of the procedures defined in the law follows:

- Every public service organization must elaborate an emergency plan (called the self-protection plan).
- The emergency plan must address all the risks associated to the activity of the organization.
- In some cases, extraordinary activities may require new sections to be added to the emergency plan.
- In other cases, some individual emergency plans can be merged into a larger one (e.g. in case of shopping centers).
- The Civil Defense Agency will set up an administrative registry of emergency plans, where all these plans will be submitted and kept for further administrative and/or operational use.
- To support the elaboration of emergency plans, the Civil Defense Agency will create and maintain a repository of emergency-related resources.
- Mechanisms for auditing emergency plans and plan maintenance will be defined and implemented.
- Online communities will be created to allow technical, administrative and managerial discussions among the different actors implied in the protection processes.
- The basic content of the emergency plan is defined by law. In some specific cases, the emergency plan must be extended with additional parts related to particular types of organizations.

- The authorities will establish certification mechanisms that guarantee that emergency plans are compliant with regulations.

Any solution for improving emergency plan management must include the above services. From a wider perspective, a number of features must characterize the solution:

Full lifecycle support. Any emergency plan management framework must provide utilities for all the actors involved in their management, from emergency plan designers to auditors, from responders to citizens participating in drills or actually involved in an emergency. Among others, emergency plan authoring tools, a emergency plan registry, emergency plan audit tools, simulation environments, and emergency plan enactment services must be included. The framework should implement the different workflows plan management is composed of, such as emergency plan submission, emergency plan audit process, emergency plan training exercise, and others.

Extensibility. Current emergency plans are documents whose basic structure and content are prescribed by local, regional or nation-wide regulations. Typically, safety enforcing laws include emergency plan templates to be used by emergency plan designers (NBA, 2007; FEMA, 2008). In the Spanish case, a basic template is defined for every organization; additional domain specific templates are defined to be plugged into the basic one in cases like nuclear plants, airport, and a number of strategic infrastructures.

Content reuse. While every organization has to define its own emergency plan, customized to its nature and activities, there are significant parts of emergency plans that are common to distinct organizations; this is the case of preambles, technical information about safety equipment, or law fragments included in the emergency plan. Moreover, a specific risk should be managed by different organizations following similar procedures.

Process awareness. At the core of an emergency plan are the activities to be performed by the different response teams to mitigate the effects of an incident. Such activities are not isolated; rather, they are part of an overall choreography where different roles are played by individuals or teams, under the direction of decision makers at the control room. At the USA, the National Response Framework is defined in terms of roles, activities, and other related concepts (FEMA, 2008). However, no specific language to define the process is provided. A formal definition of the process that such choreography defines is mandatory to lead response one step farther, and has several advantages: first, a process definition language sets a common terminology for every emergency plan; second, the formal semantics of the language, and the existence of execution environments, facilitate the analysis of the emergency plans, the definition of scenarios for teams training and, most important, the execution of the process during the response (providing coordination, data flow, efficiency, logging, etc.).

Tool support. The above requirements are hard to fulfill without tool support. Such tools should be provided by agencies, so that organizations can have a means to develop emergency plans according to the regulations and agencies can define emergency plan analysis mechanisms.

THE ARCHITECTURE OF SAGA

In this section, we describe our proposal to fulfill the requirements stated above. SAGA is a framework that integrates a suite of tools supporting the different stages of the emergency plan lifecycle (creation, audit, simulation, etc.). Related work in emergency management focus on specific aspects such as the definition of plans, for example, WILBER (Watkins, Simon-Agolory, Venkateswaran and Nam, 2011), a system to automatically generate information based on localized threats within a geographical area to extend a basic disaster preparedness plan for individuals and families. WILBER combines current and historical information from Geographical Information Systems (GIS), risk assessment, wireless sensors, and computing. Other proposals focus on the requirements capture for specific emergencies, for example, the GDIA approach (Prasanna, Yang and King, 2009), a protocol for capturing information requirements in fire emergencies. In the response emergency field, there are some proposals, such as DERMIS (Turoff, Chumer, Van de Walle and Yao, 2004), an interorganizacional Dynamic Emergency Response Management Information System. DERMIS is a transaction system integrated with a structured group communication system that can be used for all phases of the emergency response process; roles and event templates can be created and modified at any time, e.g. the system can be evolved by the users; other proposal is WIPER (Pawling, Schoenharl, Yan, Madey, 2008), an emergency management system to detects possible emergencies form cellular communication data, attempts to predict the development of emergency situations, and provides tools for evaluating

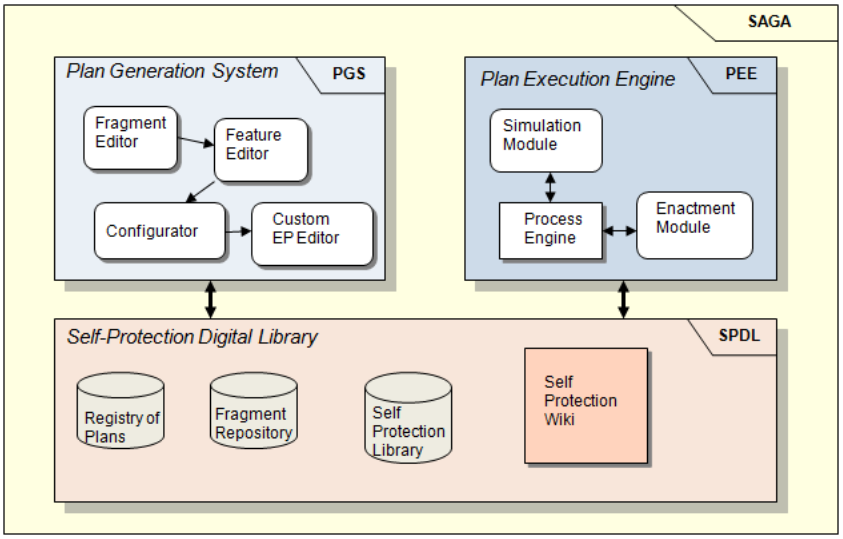


Figure1. Architecture of SAGA

possible courses of action in dealing with emergency situations ; or the proposal of (Sell and Braun, 2009) to use a Workflow Management System to execute o manage emergency plans.

In our proposal, the emergency plan is the main block and guides the rest of stages, from planning to response or simulation. Therefore, the main difference is that SAGA integrates all the stages into an architectural proposal. Figure 1 shows the overall architecture of SAGA, which we describe in detail in the forthcoming sections.

The Self-Protection Digital Library

At the core of SAGA, the Self-Protection Digital Library (SPDL) acts as the main knowledge resource. It is composed of several repositories (see Figure 1). First, the Registry of Plans holds the collection of emergency plans that organizations submit following the legal requirements. Due to its relevance from the management point of view, the Registry of Plans is described in more detail in following sub-section. Second, the Fragment Repository holds the components that are (re) used during plan development using the Plan Generation System, described later in this paper. Third, the Self-Protection Library is a collection of documents that might help emergency plan designers in their tasks; legal documents, technical information about safety equipment, or sample emergency plans may be part of the collection, which is typically maintained by the Civil Defense Agency. Finally, other components may be included in the Self-Protection Digital Library; an example is the Self-Protection Wiki, a collaborative space where practitioners can share experiences and knowledge about emergency plan development and use.

The Self-Protection Digital Library provides persistence to the emergency plans and the documents stored. Additionally, retrieval mechanisms are available for every type of object in the Digital Library, in both metadata-based and content-based modalities. There are different interfaces to the Digital Library. On one hand, a Web-based interface allows users to browse and/or search the different collections; as expected, different users of the system have access to different sections of the library. On the other hand, a service-oriented interface provides access to other components of the framework (see section “The Plan Execution Engine”).

The Registry of Plans

The Registry of Plans is the administrative entity that holds a record of all the existing emergency plans; it is a repository where emergency plans are stored in two formats. On one hand, those emergency plans developed before using SAGA are stored as PDF files. Emergency Plans developed using the SAGA Plan Generation System (see section The Plan Generation System) are stored in the internal plan format, an XML document compliant with the emergency plan metamodel of Figure 2. The metamodel is derived from the specification of emergency plan content that appears in the self-protection law.

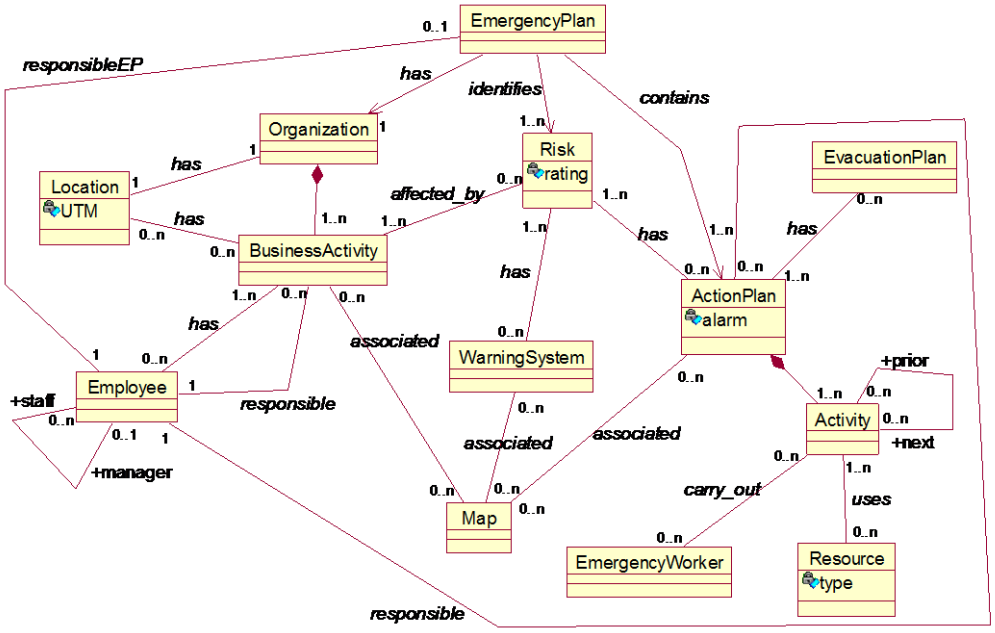


Figure 2. Emergency plan metamodel as defined in (Penadés et al., 2011)

Besides their content, emergency plans are decorated with both descriptive and technical metadata. For practical reasons, especially to preserve compatibility with non SAGA-born plans, metadata are stored separately of plans, although they are both integrated at a conceptual level (following the model of digital object services proposed in (Kahn and Wilensky, 1995)).

Several workflows are defined within the Registry of Plans. Specifically, for non-SAGA emergency plans, organizations must fill-in the plan metadata form, followed by the upload of the PDF file containing the plan. Next, the civil defense agency validates the plan, which is added to the Registry of Plans if it is compliant with regulations. Otherwise, the organization owning the emergency plan is warned and requested to make the necessary improvements. In the case of SAGA-born emergency plans, several control checks are included in the Plan Generation System, reducing registering time and complexity.

The nature of self-protection management in Spain makes the Registry of Plans a distributed entity, since every regional government owns and maintains the Registry of Plans corresponding to organizations in their geographic coverage. Thus, in some cases, especially in emergency responses, interoperability mechanisms must be defined to ensure the fluid information exchange between the local Registries of Plans and the central civil defense agency.

The Plan Generation System

Developing emergency plans may involve a high degree of reuse, but at the same time a high variability since a common strategy to respond to specific risks must be customized to the specific needs of an organization. The basic structure and minimal content of emergency plans are prescribed by law, so that, in general, emergency plans of organizations of the same domain share a high amount of generic content, being the variable parts customizations of pre-defined templates. As a consequence, emergency plan designers must be provided with tools that allow them customize templates from sets of common contents.

SAGA provides support to emergency plan design and development. Specifically, the Plan Generation System is an implementation of the Document Product Lines (DPL) proposal for the case of emergency plans, as defined by (Penadés, Canós, Borges and Vivacqua, 2011). DPL was created with a twofold goal: on one hand, to make variable content document affordable to non-expert users by including a domain engineering process previous to the document engineering one; on the other hand, to enforce content reuse at domain level following principles of software product line engineering (Penadés, Canós and Borges, 2010).

DPL provides methodological guidelines to model the commonality and variability in a document family as a set of features. Each feature is associated to a document fragment, which will be placed in the right place in the document from a fragment repository supporting the product line. The set of features selected for a specific document will be used to generate a customize editor, which is similar to a *wizard*, that will be used to create and add content to the document. The key for the success of a DPL process relies on the definition of good feature models and on the existence of an organized collection of document fragments, as well as of the appropriate mechanisms for their creation and management.

Figure 3 shows the main elements which take part in the Plan Generation Systems. The generation of an emergency plan is done in three stages, as proposed by DPL. In the Domain Engineering phase, a characterization of the global domain is done. A *domain engineer* (an expert in the emergency management field) uses the *feature editor* to characterize the variability of the domain, taking into account the types of organization and their associated risks. Each feature represents a part of the emergency plan content and it is associated to a document fragment, which may already exist in the *Fragment repository* for its reuse. If no suitable fragment is found at the repository for a given feature, the *Fragment editor* is used to create it and to add it to the repository. The feature definition stage finishes when all document fragment exists in the *Fragment Repository*. The final result of this stage is a *document template* which includes every possible component an emergency plan may have.

In the second stage, called Document Engineering, the global template is customized to a specific organization. To achieve this goal, a *document engineer* (the person in charge of creating a specific plan) selects the features that he/she considers the emergency plan must include using the *configuration editor*. As a result, a customized model is obtained and used to generate a customized editor for the emergency plan. Finally, the customized emergency plan editor is used to fill the different sections of the emergency plan that require user-provided data. Since parts of the emergency plan will be reused, the editor will only request (generally via forms) the user the information required to generate the final emergency plan.

Once the document engineer has finished the editing tasks, a fully instantiated emergency plan is generated, which is then registered at the Registry of Plans and stored in the Self-Protection Digital Library.

The Plan Execution Engine

One of the advantages of using the SAGA Plan Generation System to develop emergency plans is the possibility of defining the response procedures using more precise alternatives to natural language. In section "Requirements for a Solution", we discussed the advantages of using formal process languages to describe response procedures, being the most remarkable the ability of automatically enacting those procedures. Summarizing, a process is composed of

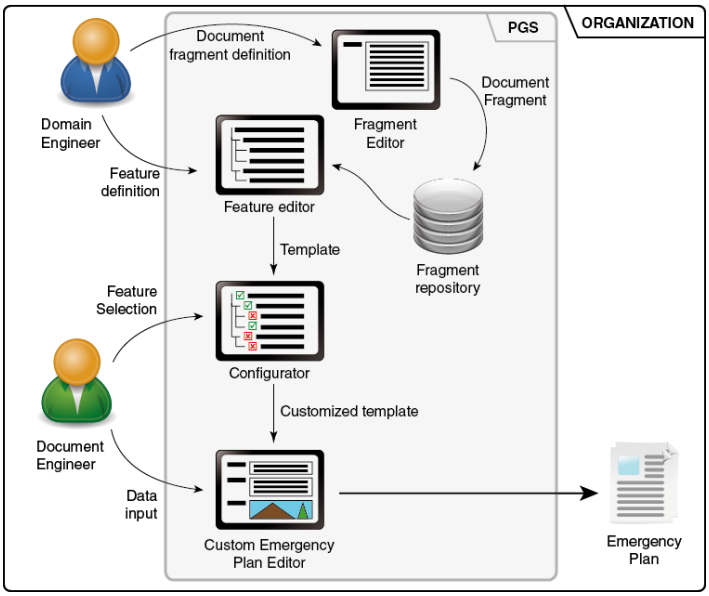


Figure 3. Main Architecture of the Plan Generation System (PGS)

a series of tasks or activities performed by different actors, in a given order that may be sequential or concurrent; the ordering of tasks is called the control flow. Additionally, data flow through the process in the form of input and output parameters of the tasks. In most process languages, a process model is a directed graph with a special starting node, and one or more final nodes. As a consequence, and depending on the value of some process variables, the graph may be traversed through different paths.

The Execution Engine is the component of SAGA that provides executability to response procedures described using process models. The process language chosen for the SAGA Plan Generation System is BPMN. However, the DPL approach would allow defining response procedures using other languages, such as the one defined in (Llavador, Letelier, Penadés, Canós, Borges, and Solis, 2006) for the specification of flexible processes.

The Execution Engine module implements the operational semantics of the language, generating the control and data flow defined in the procedure specification, as well as the assignment of tasks to the different participants in the response. Besides coordination, the actual process execution can be logged for further analysis and eventual improvement. The Engine can work in two modes, namely Simulation and Response. The former is interesting to support what-if reasoning, training teams, of just to check the correctness of the plan. The latter corresponds to the use of the plan in actual emergencies, and has stronger requirements.

The Simulation Module loads an emergency plan from the Self-Protection Digital Library, processes its procedure specification, and shows a graphical representation of it, allowing the user to define a simulation scenario, that is, a specific environment or context and a specific path along the graph. Then, the Engine starts the enactment of the scenario, offering the typical functionality of simulators (e.g. event occurrence, changes in context, re-assignment of tasks, logging …). Unlike simulations, the actual occurrence of an emergency requires to setup a hardware and software infrastructure supporting the performance of all the response teams. Within this infrastructure, the Plan Enactment Module is the main software component that provides coordination, communication, and personalized access to information to all the participants.

IMPLEMENTING SAGA

SAGA is an architectural framework that can be instantiated in different ways. The extensibility and configurability requirements made us to choose one of the state of the art development environments, namely Eclipse, for creating its first version. SAGA has been built using two key technologies: the Equinox framework (McAffer, VanderLei and Archer, 2010) and the Eclipse Modeling Framework (EMF) (Steinberg, Budinsky, Paternostro and Merks, 2009). Equinox is an implementation of the OSGi standard (OSGi Alliance, 2009), a dynamic component model and a service platform to build modular and extensible Java applications. EMF is a framework to build applications using the Model–Drive Engineering (MDE) paradigm, raising the level of abstraction and reducing development time by using code generation techniques. SAGA is made up of the three main components mentioned above, connected following a client–server architecture.

The Self-Protection Digital Library provides a centralized service for information storage and retrieval. Any kind of digital object, regardless of its type and structure, can be stored. Examples of digital objects are document fragments (text, images, links, documents, etc.), full emergency plans, or BPMN process models (see Figure 4). Every digital object has a set of metadata attributes that can be used to search the object in the Digital Library. The Self-Protection Digital Library is accessed via a Connected Data Objects (CDO) service which provides the authentication, storage and retrieval methods, independently of the actual database management system used (PostgreSQL, in the current version of SAGA).

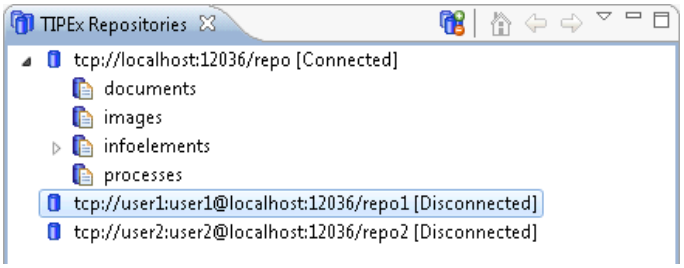
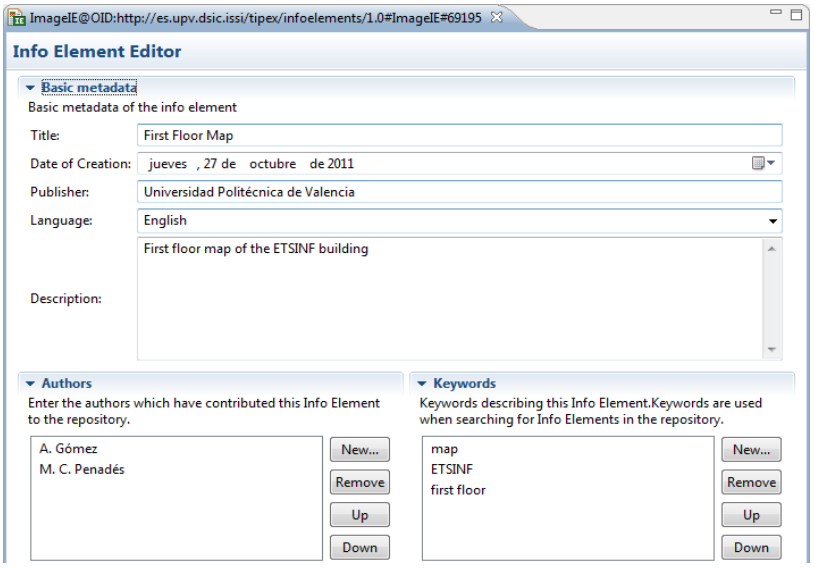


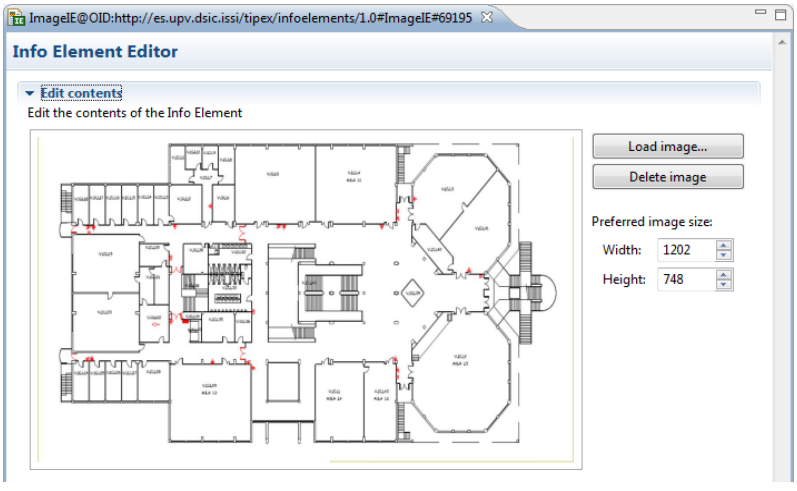
Figure 4. Self-Protection Digital Library Explorer

The Plan Generation System and the Plan Execution Engine are the client parts of the SAGA architecture and provide the different viewers and editors to define, configure, generate, simulate and enact emergency plans. They have been built as Rich Client Platform applications, and are composed of a set of pluggable components. The Plan Generation System provides support to the DPL process; it allows to explore different repositories of the Self-Protection Digital Library, create new document fragments, define document templates as feature models, create and display custom emergency plan editors and generate emergency plans.

Figure 4 and Figure 5 show the different SAGA screenshots to add new contents (an emergency plan fragment) from the Plan Generation System to a specific Repository in the Self Protection Digital Library. Figure 4 shows the *explorer*, which allows to connect to different repositories of the library. Contents in repositories are organized in a hierarchical way. When a specific location is selected, users can add any content that the Plan Generation System is able to handle. New handlers for different content types can be easily added to the Plan Generation System following the OSGi architecture. Figure 5 shows how a domain engineer edits a new document fragment; Figure 5(a) shows the metadata editor for digital objects; in this case, a new document fragment, the *First Floor Map*, is defined. The metadata editor is generic, and it is used to describe the properties of any kind of digital object. Finally, Figure



(a) Defining metadata for a plan fragment



(b) Assigning content to a plan fragment

Figure 5. Editing a new document fragment

5(b) shows the content editor, with which the document engineer assigns content to the new document fragment. Specifically, Figure 5(b) shows an editor to add a new image (a map) to the document fragment, *First Floor Map*. Such a map can be reused in any emergency plan designed for the building represented in it.

To validate SAGA, we are working with two real case studies, the emergency plan of a subway organization (MetroValencia) and the emergency plan of a school building (School of Computer Science of the Universitat Politècnica de València). The case studies provide us real emergency plan fragment as new contents to the Repository in the Self Protection Digital Library. Moreover, we have established a partnership with emergency planners of the Universitat Politècnica de València, to obtain feedback on the proposal, especially in the Plan Generation System. They are very interested because they have not any architectural support to planning, and the Civil Defense agencies don't provide this framework.

CONCLUSIONS AND FURTHER WORK

Emergency plans are key drivers of responses to incidents. As such, Civil Defense agencies define mechanisms for the management of emergency plans, including content specifications and administrative processes. In a highly normative context, tool support from authorities would be expected in order to standardize formats, unify analysis criteria, and make plan development easier. Similarly, administrative tools for plan management would ease the authorities' work. However, such tools hardly often go beyond the definition of forms or access grant to some mapping applications. As a result, heterogeneity in emergency plan design and lack of interoperability of the management tools are usual.

A full lifecycle approach to emergency plan management is still to come. In this paper, we have introduced SAGA, an architectural framework designed to provide support to all the stages of emergency plan development and use. SAGA covers administrative aspects of emergency plan management according to the Spanish legal regulation; it also provides support for emergency plan designers via a reuse-enforcing development environment which, additionally, promotes the use of rich formats to upgrade emergency plans expressiveness via hypermedia and formal process languages; finally, SAGA provides support to team training and emergency plan improvement via scenario-based simulations, and facilities for actual emergency plan enactment during emergencies.

At the moment of writing this paper, prototype versions of the SAGA Self-Protection Digital Library and Plan Generation System are available. We're working on the implementation of the Plan Execution Engine, with the focus put on the development of the Simulation Engine as a previous step towards the development of a full-featured emergency plan execution environment (which requires interoperability with Civil Defense systems, among others).

SAGA is an extensible framework to which new components can be added as needed. Of particular interest is the availability of tools for emergency plan improvement that complement the simulation-based tests. We are exploring the possibility of incorporating a module for emergency plan evaluation based on the Cross-Impact Analysis technique (Turoff and Bañuls, 2011) where some events will be related to the domain, while others will be directly tied to the process model we use to describe the procedures of a particular emergency plan (e.g. delays in the completion of some tasks, or resource allocation conflicts can result in changes in the process to avoid them). Other lines for future work are related to the incorporation of citizens as actors in the process, providing a personalized and context aware view of emergency plans to be accessed through people's mobile devices that will help them feel safer and find evacuation guidance, among other advantages.

ACKNOWLEDGMENTS

The work of J.H. Canós, A. Gómez and M. C. Penadés is partially funded by the Spanish *Ministerio de Educación y Ciencia* (MEC) under grant TIPEx (TIN2010-19859-C03-03). The work of M. R. S. Borges is partially supported by grants No. 560223/2010-2 and 308003/2011-0 from CNPq (Brazil) and grant No. E-26/103.076/2011 from FAPERJ (Brazil). The cooperation between the Brazilian and the Spanish research groups was partially sponsored by the CAPES/MECD Cooperation Program, Project #169/ PHB2007-0064-PC

REFERENCES

1. BPMN (2008). Business Process Modeling Notation. http://www.bpmn.org

2. Canós, J.H., Alonso, G. And Jaén, J. (2004). A multimedia approach to the efficient implementation and use of emergency plans. *IEEE Multimedia*, 11(3): 106-110.
3. Catarci, T., Leoni, M., Marrella, A., Mecella, M., Russo, A., Bortenschlager, M. and Steinmann, R. (2011) WORKPAD: Process Management and Geo-Collaboration Help Disaster Response. *International Journal of Information Systems for Crisis Response and Management (IJISCRAM),* Volume 3, Issue 1, pp. 32-49.
4. CDO, Connected Data Objects. http://www.eclipse.org/cdo/
5. FEMA's National Response Framework (2008). http://www.fema.gov/emergency/nrf/
6. Galton AP, Worboys M. (2011) An Ontology of Information for Emergency Management, *8th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2011)*, Lisbon, Portugal.
7. Kahn, R. and Wilensky, R. (1995). A Framework for Distributed Digital Object Services. <<http://www.cnri.reston.va.us/k-v.html>>
8. Llavador, M. Letelier, P., Penadés, M.C., Canós, J.H., Borges, M., Solís, C. (2006). Precise yet Flexible Specification of Emergency Resolution Procedures. *Proc. of the 3rd International Conference on Information Systems for Crisis Response and Management (ISCRAM'2006)*, pp. 110-120. Newark (NJ), USA.
9. McAffer, J and Lemieux, Jean-Michel. (2005). Eclipse Rich Client Platform: Designing, Coding, and Packaging Java(Tm) Applications. Addison-Wesley Professional. ISBN:0321334612.
10. McAffer, J., VanderLei, P. and Archer, S. (2010). OSGi and Equinox: Creating Highly Modular Java Systems. Eclipse series. Addison-Wesley. ISBN: 978-0321585714.
11. NBA: Norma Básica de Autoprotección (Spain's Self-protection Law) (2007). www.boe.es/boe/dia/2007/03/24 / pdfs/A12841-12850.pdf
12. OSGi Alliance. OSGi Service Platform Core Specification (2009). http://www.osgi.org/Specifications/HomePage/
13. Pawling, A., Schoenharl, T., Yan, P., Madey, G. (2008) WIPER: An Emergency Response System. *Proceedings of the 5th International ISCRAM Conference – Washington, DC, USA.*
14. Penadés, M.C., Canós, J.H. and Borges, M. (2010). Document Product Lines: Variability-driven Document Generation. *Proceedings of the 10th ACM Symposium on Document Engineering (DocEng2010).* Manchester, UK.
15. Penadés, M. C, Canós, J. H., Borges, M. R. S. and Vivacqua, A. S. (2011). A Product Line Approach to the Development of Advanced Emergency Plans. *Proc. of the 8th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2011)*. Lisbon, Portugal.
16. Prasanna, R., Yang, L. and King, M. (2009). GDIA: a Cognitive Task Analysis Protocol to Capture the Information Requirements of Emergency First Responders. *Proceedings of the 6th International ISCRAM Conference,* Gothenburg, Sweden.
17. Sell, C. And Braun, I. (2009) Using a Workflow Management System to Manage Emergency Plans. *Proceedings of the 6th International ISCRAM Conference – Gothenburg, Sweden.*
18. Steinberg D.; Budinsky, F.; Paternostro, M. & Merks, E. (2009). EMF: Eclipse Modeling Framework. Eclipse series. Addison-Wesley. ISBN: 978-0321331885.
19. Turoff M., Chumer, M., Van de Walle, B.A. and Yao, X. (2004) The Design of a Dynamic Emergency Response Management Information System (DERMIS), *Journal of Information Technology Theory and Application* (JITTA), June, 5:4, pp.1-36.
20. Turoff M. and Bañuls, V. (2011). Major Extensions to Cross-Impact Analysis. *Proc. of the 8th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2011)*. Lisbon, Portugal.
21. *Watkins*, K., Simon-Agolory, K., Venkateswaran, A., Nam, D. Get A Plan! Automatically Generating Disaster Preparednes Plans Using WILBER. *Proc. of the 8th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2011)*. Lisbon, Portugal
22. White, Connie M. (2011). Social Media, Crisis Communications and Emergency Management: Leveraging Web 2.0 Technology. Frances and Taylor Publications, CRC Press.