Turning Emergency Plans into Executable Artifacts

José H. Canós-Cerdá, Juan Sánchez-Díaz, Vicent Orts, Mª Carmen Penadés
ISSI-DSIC
Universitat Politècnica de València, Spain
{jhcanos|jsanchez|mpenades}@dsic.upv.es

Abel Gómez
AtlanMod
École des Mines de Nantes – INRIA – LINA
abel.gomez-llana@inria.fr

Marcos R. S. Borges
GRECO-PPGI
Universidade Federal do Rio de Janeiro
mborges@nce.ufrj.br

ABSTRACT

On the way to the improvement of Emergency Plans, we show how a structured specification of the response procedures allows transforming static plans into dynamic, executable entities that can drive the way different actors participate in crisis responses. Additionally, the execution of plans requires the definition of information access mechanisms allowing execution engines to provide an actor with all the information resources he or she needs to accomplish a response task. We describe work in progress to improve the SAGA's Plan definition Module and Plan Execution Engine to support information-rich plan execution.

Keywords

Emergency Plan, Knowledge Intensive Workflow, Digital Object Architecture

INTRODUCTION AND MOTIVATION

When a disaster occurs, the response procedure is performed according the specification contained in the emergency plan. Such a specification includes the identification of the actors participating in the response, plus the activities each actor should perform. These activities are modeled as tasks and coordinated according to a given control flow that defines a task execution ordering. The control flow specification may be implicit in a natural language based narrative description, or it may be explicit in the form of a directed graph in the case of formal or structured models (e.g. Petri Nets or BPMN specifications, respectively). Explicit models are mandatory to have a formal execution semantics allowing the enactment of the response procedures with the help of a process support system.

Executable plans are the natural step forward in emergency response and management. Some authors have explored the use of formal process languages to model responses (Llavador, Letelier, Penadés, Canós, Borges and Solís, 2006; Sell and Braun, 2009; Hoffmann, Sackmann and Betke, 2013). All these proposals show how a process engine can orchestrate the response processes so that coordination and logging are enforced. The type of workflow to be defined is not critical in efficient execution, but on two requirements: flexibility at runtime and access to all the knowledge required to perform a given action. Most of the knowledge is formal, and must be included in the plan itself. However, a workflow definition language is not rich enough to define a knowledge intensive environment. Moreover, it would require model emergency plans as workflows, which is not the exact view. Rather, we see a plan as an artifact that includes a workflow that accesses – among other resources – other parts of the plan. Providing access to these resources is among the main challenges of making plans executable.

In this paper, we describe how the information can be provided to the actors participating in a response process via the process support system. We show how we can specify and handle all the information resources needed by a responder to perform a given task using the so-called Knowledge Intensive Workflow models (Papavassiliou, Ntioudis, Abecker, and Mentzas, 2003). We also describe the enhancements we are giving to the SAGA framework (Canós, Borges, Penadés, Gómez and Llavador, 2013) in order to transform emergency plans into executable, knowledge intensive processes that can be enacted and monitored using a process engine. We have extended the functionality of the *Plan Generation System* with a new emergency plan metamodel that includes explicit modelling of the informational needs of the response tasks. We have also modified the *Plan Execution System* to provide an information rich execution environment.

This paper is structured as follows. First, we provide some background on SAGA and knowledge-intensive Workflows. Next, we describe how we switched from basic plans to executable response models and how it is being added to SAGA. An outline of the further work concludes the paper.

BACKGROUND

In this section we provide context for our work. First, we recall the main features of the SAGA system, and next we briefly introduce the so-called knowledge intensive workflows that will provide the conceptual base for the executability of plans.

Managing Emergency Plans with SAGA

SAGA (Canós et al., 2013) is a framework for emergency plan management that was conceived and developed to provide emergency planners with a tool supporting all the emergency plan lifecycle activities. Figure 1 shows the modular architecture of SAGA. At the core of the system, the *SAGA Digital Library* (SDL) acts as the knowledge repository where emergency plans are stored, along with other knowledge resources related to the field. Also, the SDL includes a fragment repository used by the *Plan Generation System* (PGS). The PGS 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 provides methodological guidelines to model the commonality and variability in a family of emergency plans as a set of features. Each feature is associated to a document fragment or InfoElement, which can be of different types (e.g. text, image, video). An emergency plan is generated from a set of InfoElements following a product-line strategy.

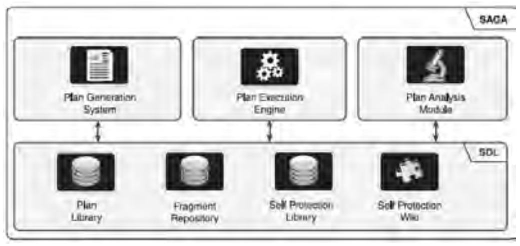


Fig. 1. Architecture of SAGA

Two additional modules complete the architecture of SAGA. The *Plan Execution Engine* supports the enactment of response procedures specified as formal process models, and is the main focus of this paper. On the other hand, the *Plan Analysis Module* is intended to perform plan analyses and suggest improvements to plan authors following different techniques. In this paper, we describe the internals of the *Plan Execution Engine*, and how we adapted the content of plans to provide executability.

Knowledge intensive workflows

The scenarios associated to Emergency Management Systems are characterized by being highly dynamic and subject to eventualities and exceptions. These characteristics introduce new requirements to classical workflow management systems. A Knowledge-intensive Workflow contains activities and tasks that can change at runtime depending on the execution context of the process (e.g. availability of resources or information needed to accomplish a task). In (Papavassiliou et al., 2003) the authors introduce a tool supporting knowledge-intensive workflows by integrating knowledge-related tasks into business processes. Two kinds of tasks are involved on the proposal: regular tasks, which describe the structured work in a process, and knowledge management tasks, which describe the generation and distribution of knowledge in a business process. The tool employs a workflow engine for process execution, and proactively delivers context-sensitive knowledge to the users for performing their work.

The proposal has been applied to public administration processes. Our aim is to apply a similar approach to the case of emergency response processes. On one hand, we want to provide operational semantics to the response procedures, so we can orchestrate and monitor responses; on the other hand, we want to provide responders with a tool to access to all the information required to perform a specific response task adequately, taking into account that most of it will come from the emergency plan. This requires some infrastructure to be added to our current SAGA plan management framework.

A ROADMAP TO THE EXECUTABILITY OF PLANS

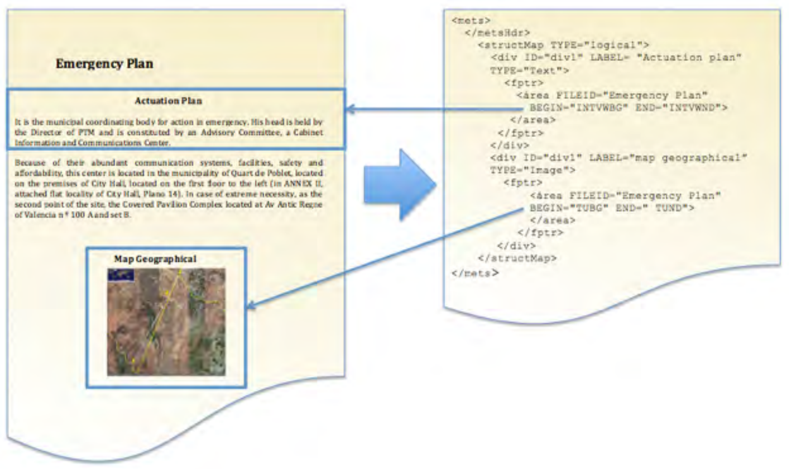
Often, a responder may need to access to some information he or she needs to perform a response task adequately. Such information is typically included in the emergency plan, but it can also be elsewhere and has to be retrieved using some search engine. In manually executed responses, the only tool that a responder needs to access to content is one allowing to browse the emergency plan content and/or perform content searches either on some search engine or a known repository. In the case of executable plans, things are a little more complicated, since the knowledge pieces must be accessed from the process engine runtime system. This requires a number of features not included in text based emergency plans that we describe below.

First, and most important, an emergency plan must be an entity manageable by a process engine. Manageability consists, at least, of having an identifier and a public interface to allow external access. The interface must offer a structure map that can guide manual browsing, plus a path definition language that allows referencing to specific parts of the plan from the specification of a task in the response process model. Second, the process metamodel used to define formal response procedures must be aware of structured plans to allow associate parts of a plan to specific tasks in the process at modeling time. And third, the runtime system must be able to traverse such associations to provide all the resources to the tasks in the process.

From plain documents to structured objects

The first step of the transformation of text-based plans into executable artifacts is the enrichment of the information environment the *Plan Execution Engine* will work upon. In the Web (and Web services) age, we must find a way to make all knowledge items available through some Web-based infrastructure. Our proposal, described later in this paper, is based on the Digital Object Architecture or DOA (Dening and Kahn, 2010). The DOA is based on four key principles: first, units of information of any type may be structured as Digital Objects (DOs) described via some metadata. Second, every DO has a unique persistent identifier, called a digital object identifier, which can distinguish a DO (or separately identified parts of it) from every other object, present, past or future; an identifier resolution system such as Handle (www.handle.net) or DOI (www.doi.org) must be available to transform identifiers into actual locations where DOs are stored. Third, DOs can be stored in DO Repositories, which, in response to a request for a DO, produce a dissemination of the DO. And fourth, accesses to an instance of DO Repository are made via a standard DO protocol called Digital Object Protocol (DOP) (CNRI, 2010); the protocol is accessible via an Application Public Interface (API).

The DOA can be implemented in different platforms, since it was designed to be neutral with respect to technology. To bring the DOA into reality, the concept of DO must be reified in some structured entity able to package the DO content along with its metadata. To achieve this, we have chosen the Metadata Encoding and Transmission Standard (METS, http://www.loc.gov/standards/mets/), developed by the Library of Congress to support the XML-based packaging of different types of metadata associated to a DO to facilitate its transmission, among other tasks. A METS package is composed of a number of sections aimed at collecting different properties of the DO. Besides the metadata records associated to a given DO, two other sections are important. On one hand, the file section records all the file(s) that comprise the actual DO content. It is organized in file groups, created according to the criteria of the DO's owner. A file usually refers to external content (by means of a URI), although content can also be embedded in the METS document as binary or XML data. Files can be linked to both administrative and descriptive metadata to establish access rights or



It is the municipal coordinating body for action in emergency. His head is held by the Director of PTM and is constituted by an Advisory Committee, a Cabinet Information and Communications Center.

Because of their abundant communication systems, facilities, safety and affordability, this center is located in the municipality of Quart de Poblet, located on the premises of City Hall, located on the first floor to the left (in ANNEX II, attached flat locality of City Hall, Plano 14). In case of extreme necessity, as the second point of the site, the Covered Pavilion Complex located at Av Antic Regne of Valencia n° 100 A and set B.

```
<mets>
  </metsHdr>
    <structMap TYPE="logical">
      <div ID="div1" LABEL= "Actuation plan"
      TYPE="Text">
        <fptr>
          <área FILEID="Emergency Plan"
            BEGIN="INTVWBG" END="INTVWND">
          </area>
        </fptr>
      </div>
      <div ID="div1" LABEL="map geographical"
      TYPE="Image">
        <fptr>
            <área FILEID="Emergency Plan"
            BEGIN="TUBG" END=" TUND">
            </area>
          </fptr>
        </div>
    </structMap>
</mets>
```

Fig. 2. Representing Plans as METS packages. The document is split into fragments stored in files that are referenced from the structural map of the METS document.

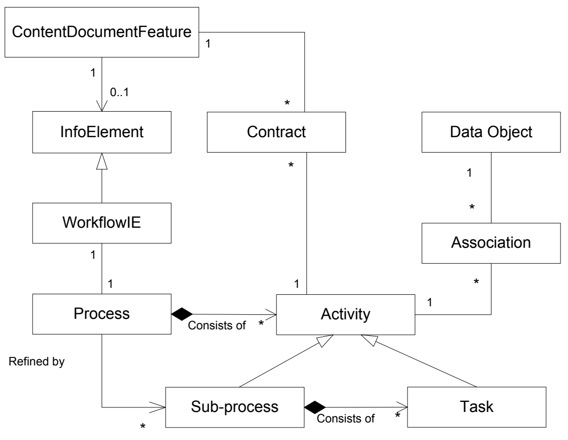


Fig. 3. Extension of the DPL Metamodel to include Knowledge Intensive Processes

other properties. On the other hand, the structural map provides structural metadata describing the structure of the DO as a set of divisions where content is placed. A division includes one or more references to files declared in the file section. Additionally, divisions are organized hierarchically via composition of divisions. Finally, references to the different parts of a METS object are declared in the structural link section.

We are extending DPL to generate emergency plans as METS packages. This way, we provide the *Plan Execution Engine* with the navigational facilities that are needed to create knowledge-intensive enactment of plans. Fig. 2 shows a partial METS representation of the emergency plan of a village near Valencia, Spain. The plan is stored in a METS XML document including the sections mentioned above.

A knowledge intensive process metamodel

The proposal of the paper is based on the idea of using knowledge intensive workflow activities to support the execution of response procedures. Fig. 3 shows at the bottom a part of the meta-model of knowledge-intensive processes we are using. It is based on the BPMN metamodel defined in (BPMN, 2004), which describes a process as composed of a number of activities that can be atomic tasks or groups of tasks called subprocesses. Such activities are connected by control flows and eventually associated with data objects. The data objects show which information is required or produced in an activity. The top part of Fig. 3 represents the extension we made to the DPL metamodel to represent knowledge intensive processes. We have added a new class named *Contract*, which represents an attachment between an activity and one resource required to complete the activity. As mentioned above, the resource will typically be a digital object representing a fragment of the emergency plan, but URLs can also be used to access Web resources. The *Contract* is the bridge between the DPL's document feature metamodel and the process metamodel we have adopted to model response procedures. A given activity will be linked to as many contracts as resources are needed for its enactment. The information contained in contract instances is used to generate the user interface for the actors participating in the emergency, as we will show later.

The *WorkflowIE* class is described as a simplified version of a BPMN process metamodel; we have included only the classes relevant for our discussion. According to BPMN, an activity is a task atomic (atomic activity) or a sub-process (complex activity). Both are assigned to roles (not shown in the Fig. 3) during workflow definition.

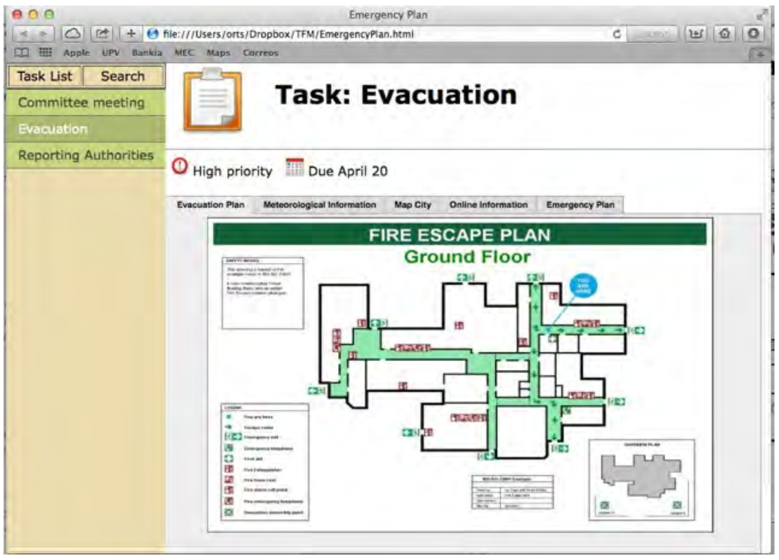


Fig. 4. Snapshot of an actor's task list during a response

IMPLEMENTING THE MODEL IN SAGA

Our work has brought the ideas underlying the DOA to the SAGA framework. We represent emergency plans as (compound) DOs, which reside in the SAGA *Plan Digital Library*, which acts as DO repository. The structure map defines navigation paths to the different parts of the plans (which are identified DOs, too). To support this, we had to change both the SAGA's *Plan Generation System* and the *Plan Execution Engine*. The former has been extended with the generation of the plan as a (compound) digital object with data, metadata, and a structure map, among other METS sections, providing access to the different components. The *Plan Execution Engine* has added digital object access in the runtime support, to allow the access to specific parts of a digital object during the execution of a task in the response process. A specific user interface is generated for every actor participating in the process, as illustrated in Fig. 4. The left frame contains the actor's task list. When the actor selects a given task from his or her task list, the main frame shows some task's details plus a tabbed set of digital object disseminations corresponding to the informational requirements of the selected task. For each contract associated to the task in the response workflow model, one tab is included in the frame. A given tab may include parts of the emergency plan or resources available elsewhere on the Web.

CONCLUSIONS AND FURTHER WORK

Making emergency plans executable is not an easy task. One needs to move from a traditional view of plans as documents towards a more sophisticated approach where a plan is a structured entity that includes different types of digital objects. Specifically, if response procedures are described in a process language, we can use the operational semantics of the language to enact plans. This is recommended not only in the context of actual responses, but also in training scenarios which can be configured via some kind of scripts.

We have brought this vision into the SAGA *Plan Execution Engine*. The existing execution environment required mechanisms to access all the information needed to perform a task. The Digital Object Architecture has provided such mechanisms in a straightforward way. We are currently finalizing the implementation of the extensions to the *Plan Generation* and *Plan Execution* modules. Further work includes the definition of scripting mechanisms to support training use of the *Plan Execution Engine*. We are also changing the way the actions in the system are logged to include information access events.

ACKNOWLEDGMENTS

This work has been funded by the Spanish "Ministerio de Economía y Competitividad" under grant TIPEx (TIN2010-19859-C03-03).

REFERENCES

1. BPMN (2004). The BPMN Metamodel. URL www.omg.org/spec/BPMN/2.0/
2. Canós, J. H., Borges, M. R. S., Penadés, M. C., Gómez, A. and Llavador, M. (2013). Improving emergency plans management with SAGA. *Technological Forecasting and Social Change,* 80, 9, 1868–1876. DOI: 10.1016/j.techfore.2013.02.014
3. CNRI (2010) Corporation for National Research Initiatives. A Brief Overview of the Digital Object Architecture and its Application to Identification, Discovery, Resolution and Access to Information in Digital Form. URL: http://hdl.handle.net/4263537/5041
4. Denning, P. J. and Kahn, R. E. (2010) The Long Quest for Universal Information Access. *Communications of the ACM*, Vol. 53 No. 12, Pages 34-36. DOI: 10.1145/1859204.1859218
5. Hoffmann, M. Sackmann, S. and Betke, H. (2013) A Novel Architecture for Disaster Response Workflow Management Systems. *Proceedings of the 10th International ISCRAM Conference – Baden-Baden, Germany.*
6. Llavador, M., Letelier, P. Penadés, M. C., Canós, J.H., Borges, M.R.S. and Solís, C. Precise yet Flexible Specification of Emergency Resolution Procedures. *Proceedings of the 3rd International ISCRAM Conference,* 110-120.
7. Papavassiliou, G., Ntioudis, S., Abecker, A. and Mentzas, G. (2003), Supporting knowledge-intensive work in public administration processes. *Knowl. Process Mgmt.,* 10: 164–174. DOI: 10.1002/kpm.176
8. 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)*. Lisboa, Portugal, May 2011.
9. Reilly, S (2009). DOP protocol Specification. Version 1.0. URL: http://hdl.handle.net/4263537/5045
10. Sell, C. and Braun, I. (2009) Using a workflow management system to manage emergency plans, *Proceedings of the 6th International ISCRAM Conference.* Gothenburg, Sweden.