A Product Line Approach to the Development of Advanced Emergency Plans

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

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

Adriana S. Vivacqua
Programa de Pós Graduação em Informática
Departamento de Ciência da Computação
Instituto de Matemática
Universidade Federal do Rio de Janeiro
avivacqua@dcc.ufrj.br

ABSTRACT

Emergency plans play a central role in emergency management processes. However, the technical problems associated to emergency plan development have received very little attention. As a matter of fact, most emergency plans are printed documents prepared with the sole support of a word processing system. As a consequence, new media are the exception in current plans. Moreover, the plans are developed without any methodological support that guides planners through the plan development process. In this paper we introduce DPL(EP), a method for the development of emergency plans. Based on the Document Product Lines process for the development of variable content document families, its main goal is to provide methodological guidance and tool support for the development of emergency plans. The distinguishing characteristics of the method are: first, the use of feature models to describe variability in emergency plan content and in the representation of the plan components; second, the "one organization-one plan" philosophy of the development process that produces customized plan editors; and third, its product line nature that enforces reuse of information elements, making plan development more convenient.

Keywords

Emergency Plan, Emergency Management, Document Product Lines, Variability Management

INTRODUCTION AND MOTIVATION

Emergency management is among the most information intensive activities ever created. To a large extent, an organization's emergency preparedness is assessed in terms of how the organization is aware of the risks inherent to its activity, and how each of the risks must be dealt with in case of incidents. In many countries, indeed, laws state that every public service organization must have its formal safety enforcement policy in an emergency plan. Such a plan must contain -or provide access to- all the knowledge required to respond to any possible emergency in the organization. Information related to the organization's infrastructure, risks associated to its activity, and procedures to respond to the different incidents, are among the most typical content of emergency plans.

The content of the emergency plans varies from one organization to another, but the aspects covered are common to all. To enforce uniformity, emergency plan structure and minimal content are set by civil defence authorities as part of the legal regulations. In (Canós, Alonso and Jaén, 2004) we analyzed the nature of emergency response, and characterized emergency information along a number of dimensions including coordination, information access and management, collaboration, and others. Thus, any emergency plan has parts related to each of these dimensions. Additionally, some general data such as the organization details

Reviewing Statement: This full paper has been fully double-blind peer reviewed for clarity, relevance, significance, validity and originality.

(name, activities, location, infrastructure, etc.) are included in the emergency plan.

The response to a specific emergency is, actually, the enactment of its corresponding part of the emergency plan. In a response, the intervention of the different teams is coordinated following the directives included in the plan. Given the diversity of people and organizations involved in a complex emergency, each one with different information requirements, providing the right information at the right time is considered a great challenge. Nowadays, however, most emergency plans are printed, text-based documents, that are used at control rooms to transmit orders to the different response teams. In some cases, this centralized view is a source of bottlenecks that may affect the performance of responders. To streamline responses, the newest Information and Communication Technologies can be used.

Several research projects have focused on increasing the use of Information and Communication Technologies in emergency management, especially during emergency responses (Security Research, 2009). Specifically, the use of mobile devices allows the transmission of multimedia information from control rooms to the emergency scenarios, providing responders with customized information. Similarly, those devices can be used to take pictures or record videos from the emergency location, and send them to the control room, increasing awareness of decision makers. To support interaction between mobile devices and servers at control rooms, different types of service oriented architectures have been proposed.

However, little attention has been paid to the problems related with the specification and elaboration of the emergency plans supporting the Information and Communication Technologies enabled responses. The distributed architecture of modern emergency management systems enables the use of advanced emergency plans, which are no longer printed, static documents, but rich behaviour applications that meet most of the response requirements. The possibility of delivering multimedia information to responders' mobile devices, in the context of emergency response workflows enactment, the monitoring of the different response activities, the support of collaborative decisions, and other features, open a spectrum of possible emergency plan implementations that must not be neglected.

Advanced emergency plan elaboration requires special skills in aspects related to Information and Communication Technologies. However, emergency plan designers, who are not necessarily Information and Communication Technologies experts, must face the construction of such plans without the adequate guidance that helps them to choose the design option that best fits their needs. As a consequence, despite the technological advances, the classical document is the dominant form of emergency plan. To avoid this, we have defined DPL(EP), a methodological approach to the definition and implementation of advanced emergency plans. The method is the instantiation of the Document Product Lines (DPL) framework to the Emergency Plans (EP) development domain. DPL(EP) supports two processes. On one hand, the specification of the content of advanced emergency plans in terms of two types of features, namely structural and technological; on the other hand, the reuse-based implementation of advanced emergency plans following a product-line strategy.

The premises DPL(EP) is based on are simple. First, separating content from layout and presentation enforces reuse of the emergency plan basic building blocks, which we will call *InfoElements*. Then, a given *InfoElement* can be represented using different models, introducing the flexibility needed to implement customized information delivery. Additionally, similar emergency plans can be built reusing components taken from a repository of *InfoElements*. Second, the emergency plan development process must fit the organizational needs, instead of obliging organizations to adapt to the emergency plan development requirements. Thus, customized plan editors may adapt to the characteristics of particular organizations, avoiding emergency planners to cope with the complexity of the overall legal regulations.

This paper is organized as follows. We first introduce our notion of emergency plan, built on top of a knowledge model that classifies information items in terms of their source. We detail the basic plan characteristics and components, giving an example based on the legal regulations in Spain. Then, we describe the DPL framework for the development of variable content documents. The adaptation of DPL to the case of advanced emergency plan development is detailed, showing how customized emergency plans can be developed from feature models.

THE NATURE OF EMERGENCY PLANS: A KNOWLEDGE MANAGEMENT PERSPECTIVE

The emergency plan is a complex document that includes the coordination mechanisms among responders, the procedures to be executed and the information to be exchanged between the different actors involved (maps, pictures, media, etc.). Most of this information is generated in advance, as a result of the prevention activities, and does not change significantly; other parts, however, are generated during the development of the emergency and complement -and sometimes override- the previously generated information. We discussed the nature of the information managed in emergency responses in (Canós, Penadés, Solís, Borges and Llavador, 2010). There, we defined a knowledge model that included the different types of information managed in emergency response. In the model (represented by the class diagram of Figure 1), the knowledge is stored in the so-called *InfoElements*, that contain meaningful information units whose origin may be the emergency plan (*FormalIE*), the emergency scenario (*ContextualIE*), or both (*CompositeIE*). Each *InfoElement* is represented as one or more digital objects (that is, text, audio, video, etc.) which can be visualized using one or more disseminators (image viewers, video players, etc.) accessible through different interfaces and/or protocols (Kahn and Wilensky, 1995).

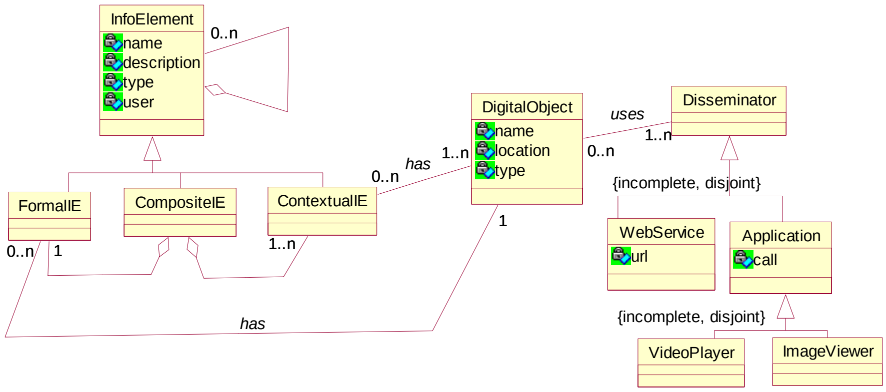


Figure 1. Knowledge Model (Canós et al., 2010).

The *InfoElement* hierarchy was used as sterotypes in the definition of emergency plan models. Each element of an emergency plan can be classified as formal, contextual or composite, so that designers have a clear specification of the requirements associated to the different parts of the plan. Figure 2 illustrated the UML model representing the content of an emergency plan based on emergency management regulations in Spain (The "Norma Básica de Autoprotección" (NBA, 2007)). The regulation specifies the (minimum) content and structure of every NBA-compliant emergency plan: information about the organization, its location, the risks of its activity, and so on. Notice that the different parts of an emergency plan are labeled using the stereotypes. This common structure must be completed with optional components for specific cases according with the NBA regulation, such as nuclear plants or airports. The model proposed in Figure 2 integrates the content and the Information & Communication Technologies in the specification of emergency plans, one of the main challenges in their development.

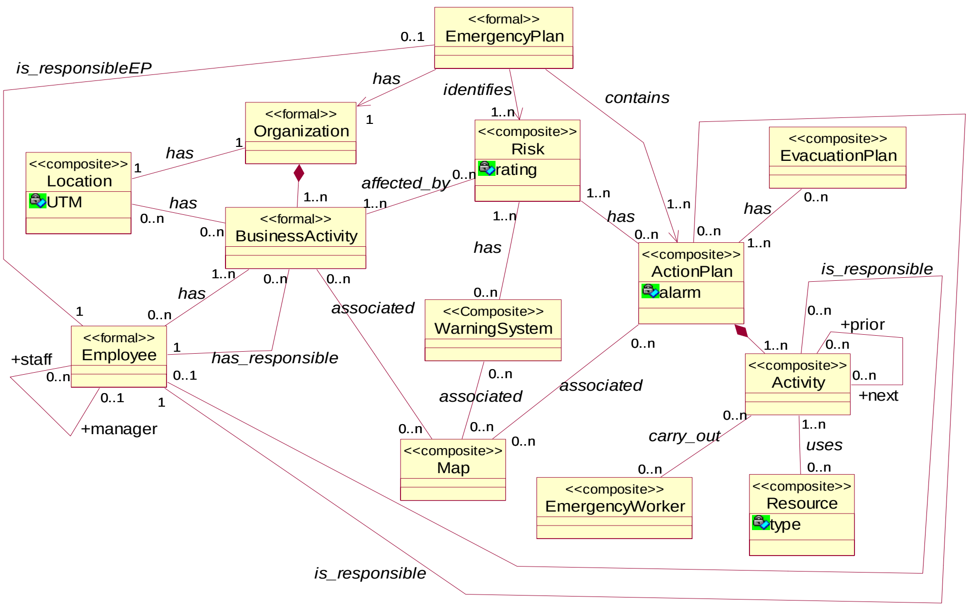


Figure 2. Emergency plan content defined in terms of *InfoElements* (Canós et al., 2010).

A tool to support emergency plan definition must be aware of the variability in content described above. But this is not the only source of variability: the technology used to represent specific *InfoElements* introduces a diversity that both emergency plan editing tools and their corresponding enactment environments must cope with. Figure 3 illustrates technology variability: the same fragment of an emergency plan is represented as plain text (left), and as a node of a hypermedia structure (right). In (Canós et al., 2004) more details about hypermedia approach to the efficient implementation and use of emergency plans in a subway organization are presented. Thus, the delivery of the adequate fragment representation will be driven by the technical capability of (the device used by) the receiver of the information.

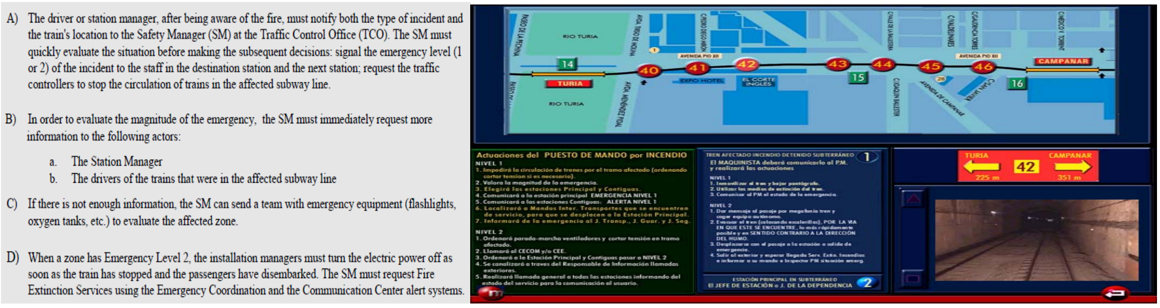


Figure 3. Technology Variability

Our method uses feature models to describe both the content and technology variability dimensions. Then, a product-line process guides the development of emergency plans with a high degree of content reuse, easing the overall planning process. In the next section, we introduce the basic principles and characteristics of the Document Product Lines process.

DOCUMENT PRODUCT LINES: A METHOD FOR THE DEVELOPMENT OF VARIABLE CONTENT DOCUMENTS

Document Product Lines (DPL) is a method for the generation of document families (Penadés, Canós and Borges, 2010). By family, we mean a set of documents that share a common set of features but that are different from each other in a number of specific features. DPL is based on principles and techniques from Software Product Line Engineering, an approach to software development defined in (Clements and Northrop, 2002) as "*a set of software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way*". DPL provides methodological guidelines to model the commonality and variability in a document family as a set of features. From the features selected for a specific document, a customized document editor will be generated using the corresponding set of *InfoElements* following a product line approach. The editor will act as a wizard that will guide the users to create and add content to the document.

DPL is composed of two iterative and incremental processes, namely Domain Engineering and Application Engineering. The goals of Domain Engineering are: first, to obtain a description of the document family in terms of content and technology features; second, to identify the so-called core assets, that is, the basic units of knowledge (the *InfoElements*) required for the development of the documents and the technological support required to edit and enact this knowledge (the *Disseminators*); and third, to define a production plan that describes the processes that guide the development of the different family documents from the core assets. On the other hand, the Application Engineering process supports the implementation of customized editors following the production plan. It begins with the document characterization phase to identify the requirements for a particular document, followed by the enactment of the production plan to obtain the customized document editor. We summarize both processes in the remainder of this section.

Describing variability using Feature Models

Feature modeling (Czarnecki and Eisenecker, 2000; Kang, Cohen, Hess, Novak and Peterson, 1990) has been used in the last decade to describe the standard characteristics that systems in a particular domain must show. In general, a feature is a user-visible aspect or property of a system or software product. Features can be classified as mandatory, optional or alternative. This way, the variability can be easily specified in terms of optional or alternative features.

In DPL, we have adapted feature models to the document domain. We define a document as the union of two components, namely content and presentation. The document content includes a template that defines the logical structure of the document, plus the data that instantiates the different elements. The presentation includes, on one hand, the layout that defines exactly where each piece of content is to be placed, and, on the other hand, the technology that is used to encode the document (e.g. printed format, windows application, web application, or multimedia application). By means of document feature models, the specification of the document family in terms of content and technology is provided. We do not consider document layout since the specific layout design techniques are orthogonal to the method.

Figure 4 shows the UML class diagram that represents the DPL's Document Feature Metamodel. The main difference with the classical feature metamodels (for an example see (Clements and Northrop, 2002; Pohl, Böcke and Linder, 2004) is the inclusion of two different types of top-level features: those related to the document content (*ContentDF*), and those related to the technology used to represent the content (*TechnologyDF*). Both types of features represent variability sources in documents, as explained in previous section. Every content feature can be composed of other content features as mandatory, optional or alternative feature (*containsCDF* association). Similarly, every technology feature also can be composed of other technology features. A key relationship of DPL feature metamodel is the *is_provided* association: every content feature must have at least one technology-related feature associated to it. Technological variability, one of the key issues in our method, means that the same *ContentDF* can be shown differently depending of the *TechnologyDF* selected. Document features may have zero or more restrictions, with "requires" being the most frequent restriction type; for instance, "*f1 requires f2*" means that a document with feature *f1* must also include feature *f2*.

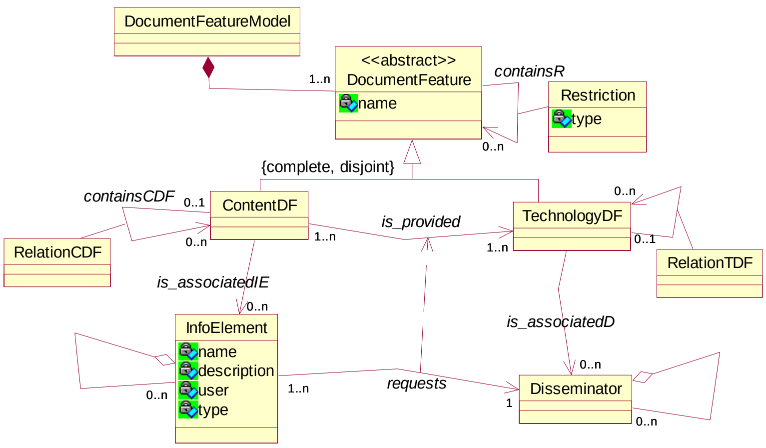


Figure 4. Document Feature Metamodel

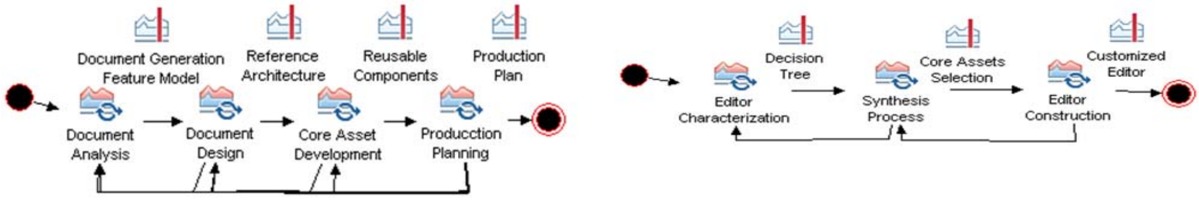
A particular document type is defined as an instance of the feature metamodel. Actual document content is associated to *ContentDFs* via instances of the *InfoElement* class, which are pieces of knowledge that are stored as typed digital objects. Moreover, the disseminators which can be used to visualize the different types of digital objects are associated to technology features.

Generating Documents from Feature Models

Figure 5 shows a more detailed view of the DPL process in SPEM notation (SPEM, 2008). The Domain Engineering process is composed of four stages or phases. In the *Document Analysis phase*, a domain engineer specifies the document features; those features composed of alternative and/or optional features are called *variability points*. In the *Document Design phase*, the customized editor architecture is designed by identifying the components to be integrated in the editor. Next, in the *Core Assets Development* phase, the core assets (*às* and *InfoElements*) are identified and validated. If no assets can satisfy the requirements specified in the feature model, new artifacts should be developed or discovered. Finally, in the *Production Planning* phase, a production plan is obtained; an important activity of this phase consists of linking content features and technology features. This linking drives the integration of the different editor components.

The Application Engineering process starts right after the Domain Engineering. In the *Editor Characterization* phase, the user selects the variability points that will guide the selection of the core assets in the *Synthesis*

Process phase. Finally, in the *Editor Construction* phase, the integration of the core assets generates the customized editor.



(a) *DPL Domain Engineering Process* **(b)** *DPL Application Engineering Process*

Figure 5. DPL Processes

INSTANTIATING DPL TO THE DEVELOPMENT OF EMERGENCY PLANS

Emergency management is a very complex domain that requires a clear understanding of the requirements and a systematic approach in order to select the variability sources before starting the emergency plan generation. However, no tools are available that guide emergency plan designers in the entire process of collecting, organizing and publishing information in an emergency plan. Being a mandatory activity, subject to supervision by civil defence agencies, it is desirable that these agencies provide supporting tools. We can find some proposals in this direction, such as GDIA (Prasanna, Yang and King, 2009) a protocol for capturing information requirements in fire emergencies. To provide supporting tools is common in other e-government contexts, such as tax statement preparation: the tax office offers a portal where citizens can fill their statement in a guided session. The fields to be filled vary according to the different taxpayer profiles, but the final result is always a document compliant with the structure and content of tax statements defined in the law.

We propose a similar approach for emergency plan development. However, being the domain more complex, an unique emergency plan document model is not affordable due to the high diversity of organizations that would result in a very large document. Thus, some more sophisticated solution is required. Our proposal is built on top of the DPL approach. First, the emergency planners apply the DPL process to obtain a family of emergency plan editors, and the functionality of which will be selected from the features of the emergency plan to develop. The emergency responsible in the organization participates in the characterization editor phase. And second, the editors are used by both participants to add content to the emergency plans. In the remainder of this section, we illustrate the application of DPL(EP) to the case of the emergency management regulations in Spain (NBA, 2007).

Generating the customized Emergency Plan Editor

We have developed a tool to support the DPL(EP) process. We have used Pure::Variants 3.0 (http://www. pure-systems.com), an open framework based on the Eclipse platform. The framework integrates several components:

- a feature model editor to represent the commonalities and variabilities of emergency plans;
- a family model editor to represent the core assets (*InfoElements* and *Disseminators*);
- a variant description model to explore the different variants in the family of emergency editors; and,
- an application developer to generate an customized emergency plan editor.

Emergency Plan Feature Model

In the *Domain Engineering* stage, we have analyzed the variability in emergency plan content introduced by the legal regulation, and built the emergency plan document feature model shown in Figure 6. The model is an instance of the metamodel of Figure 4. We use the prefix "*CF_*" in the names of the content features, to distinguish them from the technology features, whose names are prefixed by "*TF_*". Additionally, we use the exclamation mark to mark features as mandatory, question marks to mark optional features, and double-head arrow marks for alternative features. We have defined several mandatory content features (organization; risks and their warning systems; action plans and their evacuation plans). Optional content features depend on the type of organization; for instance, the *CF_Specific* feature defines the type of organization with values transport, energy, manufacturer, entertainment and other. The content features can be refined to represent the information or data that they contain until the desired level of granularity using the concepts of knowledge piece and digital object.

The overall emergency plan structure is represented as a set of *contentDF* elements. The plan specification must be completed with the specification of the variability in the technological maturity of organizations. To describe it, we have followed the multidimensional nature of emergency response defined in (Canós et al., 2004) including aspects such as coordination (how emergency resolution procedures are defined), information management and retrieval (what types of information items are used and how they are managed), their presentation to the emergency actors, the communication mechanisms, the support to collaboration between emergency responders, and finally, the use of intelligent systems for different tasks, including the extraction of knowledge from contextual data. Some of the above dimensions are explicitly reflected in the structure and content of emergency plans (for instance, the activities to be performed as a part of an action plan, are a view of the coordination), whereas others are implicit (it is clear that a robust communications system must support the emergency response processes).

The feature model shown in Figure 6 is, thus, the result of combining the multidimensional view of emergency management with the emergency plan content specification derived from law. Technology features are organized hierarchically. The topmost features are composed of other features corresponding to the different technologies that can be used to implement them. For instance, *TF_Presentation* is included because information must be provided to responders; and, depending on their technical capability, the information can be delivered either as a printed document (*TF_Printed*) or as a hypermedia application (*TF_Hypermedia*) accessed through a smartphone, to name two cases.

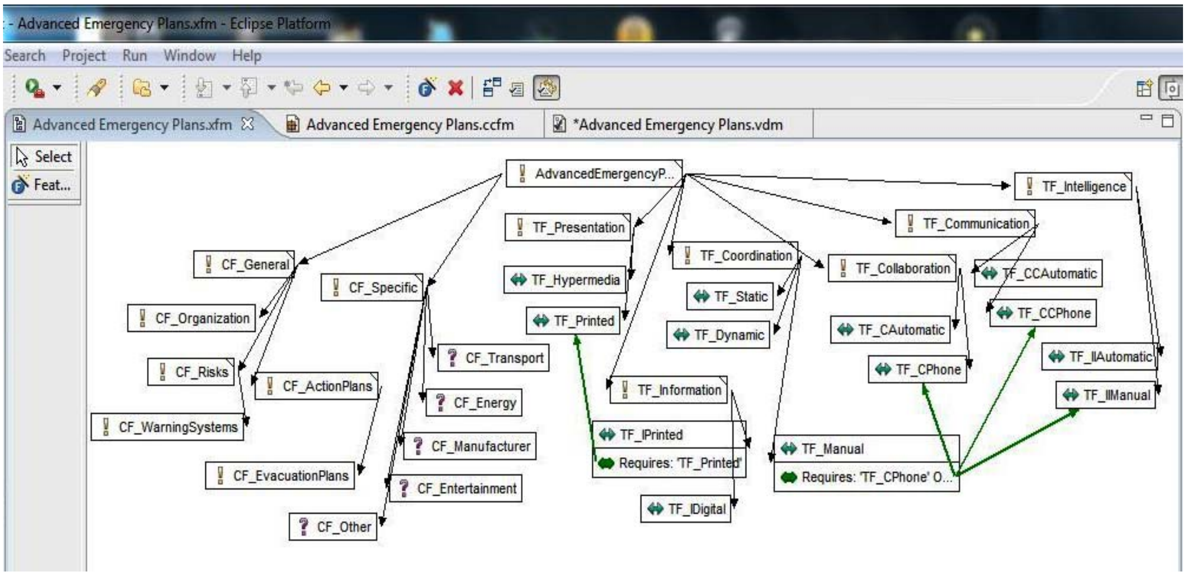


Figure 6. Emergency Plan Feature Model

Another interesting example is the *TF_Coordination* feature. It is modelled as an XOR variability point with three alternatives. *TF_Manual* feature represents the case in which response procedures or actions plans are defined using narrative texts and, as a consequence, their enactment must be made by a human actor, usually from the control room. The representation of the procedures is just a digital object (text or image, for example). The *TF_Static* feature represents the case where a formal process language is used to define the procedures, although no automatic enactment is provided. Finally, the *TF_Dynamic* feature represents the most sophisticated case of coordination, where a process language is used to define coordination, and a process engine drives its enactment. Workflow management systems would be typically associated to this feature.

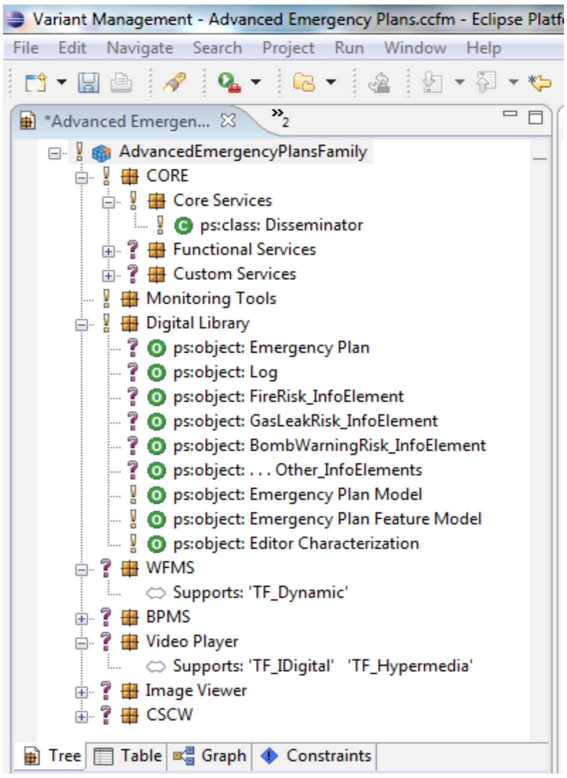
The feature model is completed with the specification of the requires relationships. For instance, if the information is mostly contained in printed documents (TF_IPrinted) then the presentation must also be printed (TF_Printed). Similarly, if the coordination is manual (TF_Manual), then the collaboration, communication and intelligence activities must also be performed manually (TF_CPhone, TF_CCPhone and TF_IManual).

Reference Architecture and Production Plan

The next steps in the Domain Engineering process are the design of the emergency plan editor and the identification of the *InfoElements* and *Disseminators* required for its development. Figure 7(a) shows all the components identified as a family model. Like in the document feature model, these components have been labelled as mandatory, alternative or optional.

The family is built around a number of components that implement the different functions of the emergency plan editors. *CoreServices* is a mandatory component that contains all the *InfoElements'* disseminator management services (e.g. add or remove a disseminator to or from the emergency plan editor). Another mandatory component is a *digital library* or digital object repository, which implements the persistence services required to store the different artifacts involved in the process, such as the emergency plan model (represented in Figure 2), the emergency plan feature model (represented in Figure 6), and the actual emergency plan developed. Additionally, the digital library component will act as the repository of the *InfoElements*, such as *Fire_Risk*, *GasLeak_Risk*, *BombWarning_Risk*, etc. Other components of the family are the disseminators associated to the *TechnologyDF* features. For instance, a workflow management system (*WFMS*) or a business process management system (*BPMS*) may be part of the customized editor when the emergency plan is including automatic enactment of the response procedures.

The Domain Engineering process concludes with the definition of the production plan. The main task in this stage consists of linking *ContentDF* and *TechnologyDF* features, i.e., to select the technologies which can be used to manage the emergency plan contents. These links are shown in Figure 7(b). For instance, the emergency plans will use technologies related to *information* management and retrieval to find information related to risks, and its *presentation* to the emergency responders. And to represent the action plans, technology related to *coordination*, *communication* and *collaboration* between emergency actors will be required.



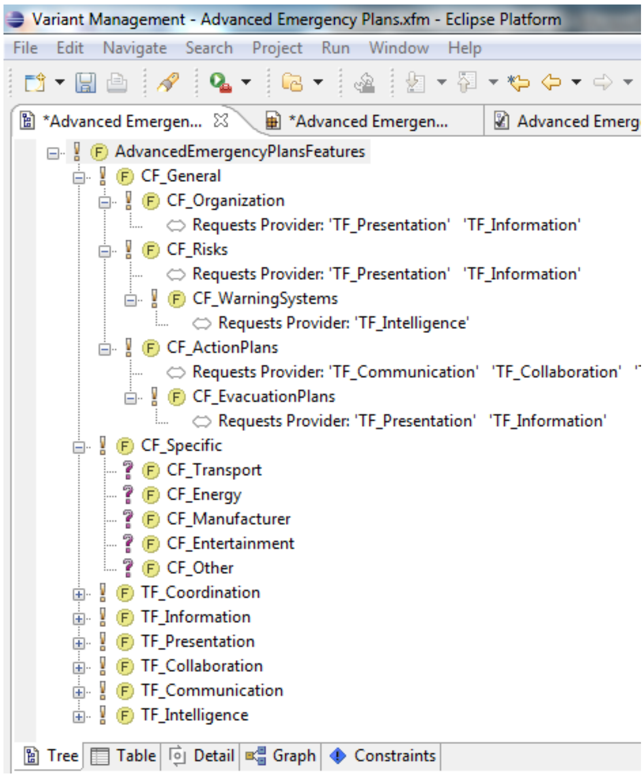


Figure 7 (a). Core Assets in Emergency Plan Editor **Figure 7 (b). Production Plan Definition**

Emergency Plan Editor Characterization

The Application Engineering process exploits the variability modelled to generate the customized emergency plan editor. The variant description model of Figure 8 is explored, and mandatory features are automatically selected, whereas the optional and alternative features that will be included in the editor are selected by the emergency plan designer in the following order: first, the emergency plan designer selects the content features by assigning the organization to one or more types included in the model. For instance, if we are generating the emergency plan of a subway organization as *MetroValencia* (Canós et al, 2010), the *CF_transport* feature must be selected. Next, the associated technology features must be selected. For instance, a hypermedia emergency plan, where the enactment is made by human actors and the information is stored as digital objects, corresponds to the following selection: `{(TF_Coordination=TF_Manual),(TFPresentation=TF_Hypermedia),(TF_Information`

`=TF_Digital),(TFCommunication=TF_CCPhone),(TF_Collaboration=TF_CPhone),(TF_ Intelligence= TF_IManual)}` From the above characterization, the customized emergency plan editor will be generated in accordance with the variability points selected.

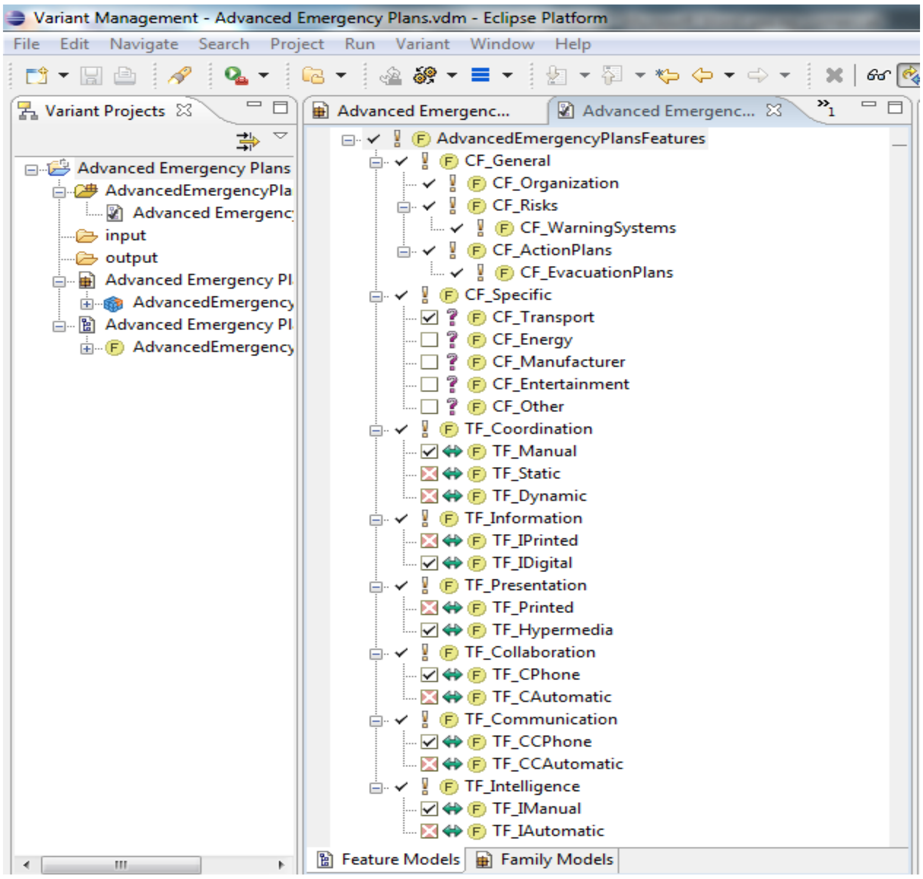


Figure 8. Example of Emergency Plan Characterization

Obtaining the Emergency Plan

The emergency plan is obtained by executing the customized plan editor, which will guide the emergency planner, following a wizard-like style, in the introduction of the plan contents, according to the structure and technology support selected. If some parts of the plan are *InfoElements* stored in the repository, and selected in the editor generation, they will be automatically added to the plan. In other cases, data will be entered by the planner; this is the typical case of the organization name, location, business activity, employees, and other properties. There are some more complex cases, like the specification of the action plans. Depending on the technology feature selected for its description, the plan editor will open the appropriate disseminator (a text editor for textual descriptions, a graphical BPMN editor (BPMN, 2008) if this is the language selected for describing the action plans, and so on.

CONCLUSION AND FURTHER WORK

The existence of an emergency plan is a requirement of every emergency management activity. The plan is a knowledge resource that assists decision makers during the resolution of incidents, and as such it must be as complete and effective as possible; this situates planning for emergencies among the most important activities of emergency management. Creating good plans requires, on one hand, a high level of expertise in the field, that will allow the risk identification, prevention and response; and, on the other hand, a good Information and Communication Technologies supporting infrastructure ensuring that the right information will reach the right actors, at the right time, and in the right format. However, the development of tools supporting emergency planning has been largely ignored as a high priority issue. In a political scenario where laws are obliging organizations to improve their plans, and in the current technological era, dominated by the emergence of new media that enable more effective ways to access to the information, the existence of emergency planning methods and their associated support tools has become a must.

In this paper, we have introduced DPL(EP), a method for the systematic development of emergency plans using Software Product Lines and Document Engineering techniques. Specifically, we have adapted a method for the development of variable content documents to the case of emergency plans. Starting from a characterization of the domain, we specify the plan in terms of mandatory and/or optional features related to plan content and the technology used to access to the content. The feature specification will be used as the requirements specification for the product line-based plan generation process. This process will build specialized document editors from a set of knowledge items or core assets that can be reused whenever two or more organizations require the same feature to be present in their plans.

We have developed a framework providing tools for the development of plans following the DPL(EP) process. The framework is defined in Eclipse, and its use has been illustrated in the last section. We are improving the framework in several directions, especially in the aspects closer to the automatic generation of the plan editors. To validate the DPL(EP) approach we are working with two real case study, 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).

ACKNOWLEDGMENTS

The work of M. C. Penadés and J. H. Canós 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 480461/2009-0 from CNPq (Brazil). The work of A. S. Vivacqua is partially supported by 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., Penadés, M.C., Solís, C., Borges and M., Llavador, M. (2010). Using spatial hypertext to Visualize Composite Knowledge in Emergency Responses. *Proceedings of the 7th International ISCRAM Conference,* Seattle, USA.
3. 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.
4. Clements, P. and Northrop, L. (2002). Software Product Lines: Practices and Patterns. Addison-Wesley
5. Czarnecki, K. and Eisenecker, U.W. (2000). Generative Programming: Principles, Techniques and Tools. Addison-Wesley
6. Kahn, R. and Wilensky, R. (1995). A Framework for Distributed Digital Object Services. <<http://www.cnri. reston.va.us/k-v.html>>
7. Kang, K., Cohen, S., Hess, J.A., Novak, W.E. and Peterson, S.A. (1990). Feature-Oriented Domain Analysis (FODA) Feasibility Study. Software Engineering Institute. Carnegie-Mellon University
8. NBA (2007). www.boe.es/boe/dia/2007/03/24/pdfs/A12841-12850.pdf
9. 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.
10. Pohl, K., Böckle, G. and van der Linden, F. (2005). Software Product Line Engineering – Foundations, Principles, and Techniques. Springer, Berlin, Heidelberg, New York
11. 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.
12. Security Research Projects under the 7th Framework Programme for Research (2009). European Commission. Enterprise and Industry. http://ec.europa.eu/enterprise/security/index_en.htm
13. SPEM (2008). Software Process Enginering Metamodel v2.0. http://www.omg.org/spec/SPEM/2.0