OntoEmergePlan: variability of emergency plans supported by a domain ontology

Maria I. G. B. Ferreira
PPGI/UFRJ – Brazil
mariaines.bosca@gmail.com

João L. R. Moreira
SCS/U. Twente – Netherlands
j.luizrebelomoreira@utwente.nl

Maria Luiza M. Campos
PPGI/UFRJ – Brazil
mluiza@ufrj.br

Bernardo F. B. Braga
NEMO/UFES – Brazil
bernardofbbraga@gmail.com

Tiago P. Sales
NEMO/UFES – Brazil
tgoprince@gmail.com

Kelli de F. Cordeiro
PPGI/UFRJ – Brazil
kelli.faria@gmail.com

Marcos R. S. Borges
PPGI/UFRJ – Brazil
mborges@nce.ufrj.br

ABSTRACT

The preparation of high quality emergency plans to guide operational decisions is an approach to mitigate the emergency management complexity. In such multidisciplinary scenario, teams with different perspectives need to collaborate towards a common goal and interact within a common understanding. In this scenario, the characterization of the variability of the elements involved in these plans is an important issue, which is addressed by the emergency plans generation methodology Document Product Line for Emergency Plans (DPL(EP)). To increase common understanding of plans, we propose an adaptation of this methodology by applying a well-founded emergency ontology, termed OntoEmergePlan, which supports the domain engineering phase. It is grounded in a foundational ontology, which ensures a higher consistency degree to the process of plans generation.

Keywords

Emergency Plan, Document Product Line, Foundational Ontology.

INTRODUCTION

An emergency management system is mainly composed of (i) human agents, such as, victims, staff and volunteers; (ii) artificial agents such as sensors, social network and information systems; (iii) material artifacts such as donatives, mobile and telecom devices, and; (iv) institutional resources such as government and humanitarian affair institutions. In this complex environment, the agents' behavior during the emergency response are partially guided by emergency plans, which describe and identify a set of characteristics, such as those responsible for carrying out specific actions, equipment, facilities, supplies, other resources available, and outline how all actions should be coordinated.

The preparation of emergency plans content faces several challenges due to the dynamic emergency environment and its variables, so the plan must be flexible enough (Boin and Hart, 2010). To overcome these challenges, Pénades et al. (2011) proposed in a prior work the methodology DPL(EP), which guides the generation of emergency plans based on variability models through feature modeling. These plans are intended to be used by different agents, both human and machines, that participate on the emergency events, thus requiring a common understanding of the involved concepts and goals. Feature models are not expressive enough to cover all the variability needed to support systems in highly dynamic environments. (Johansen et al, 2010). So, DPL(EP) does not address this requirement successfully, which can affect the plans quality and, hence, the emergency actions during the response phase.

In this paper we propose an adaptation of DPL(EP) to deal with the aforementioned limitation. We enhance the expressivity of plans contents by using a well-founded modeling language in the domain engineering phase of DPL(EP), which is fully described in (Penadés et al, 2011). This language, named OntoUML (Guizzardi, 2005), is specially designed to formalize models and is based on a set of foundational ontology categories and an ontological engineering method that aims to capture how a community classifies a portion of reality according to a prior agreement of the involved concepts. OntoUML provides highly expressive constructs and guidelines on how to use them, which improves the feature modeling activity of DPL(EP), resulting in a more consistent plan generation.

This paper is organized as follows: DPL(EP) is revised in the next section. Then, we present an overview of well-founded domain ontologies. Afterwards, we describe the DPL(EP) adaptation, firstly by presenting OntoEmergePlan and, secondly, discussing its application in the DPL(EP) method through an example scenario. We conclude this paper with our contributions and future work.

DOCUMENT PRODUCT LINE AND VARIABILITY OF EMERGENCY PLANS

The nature of emergency plans is complex and requires specific knowledge to prepare them. Emergency plans must include a set of aspects that can vary during the emergency occurrence, such as the information exchanged among the actors and the different types of technology used. Emergency plans can be considered as a document family, i.e. a set of documents that share a common set of features. A methodology to generate document families was introduced in (Penadés et al, 2011), termed Document Product Line (DPL), which is divided in modelling (domain engineering) and construction (application engineering) phases (Figure 1). It combines principles and techniques of software product line, particularly feature modeling for standard characteristics representation, and document engineering. A feature is a user-visible attribute of a software: mandatory, optional or alternative.

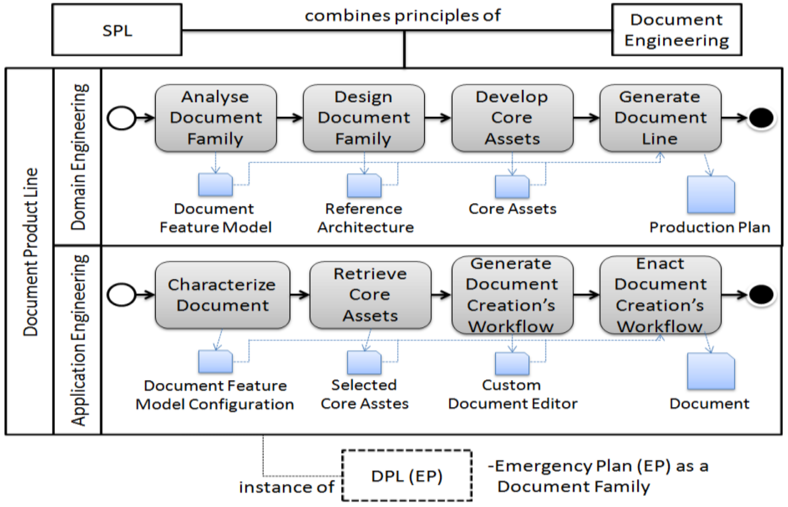

Figure 1. Document Product Line for Emergency Plans (Penadés et al, 2011)

The DPL approach provides a guideline to design the similarities and variability characteristics in a document family as a set of features. DPL represents the document family in terms of two types of features: content (information and logic structure) and presentation (layout and technology), which can be set as variability points when they are alternative, as pre-defined possibilities, or optional. The features variability definitions occur in the analysis document family activity of domain engineering phase of DPL methodology, as illustrated in Figure 1. The former supports the specification of the production plan based on: document feature model, reference architecture and essential basic unities, the core assets that can be reusable. The later supports the development of the documents editor.

The application of the DPL process in the emergency domain gave rise to DPL(EP). In this particular case, the process has an emergency plan as an instance of a document family. The goal of DPL(EP) is to provide a framework to guide emergency plan designers in extracting, organizing and delivering the adequate information in emergency plans. The emergency plan editor software is the result of the application engineering phase of DPL(EP) method. Then, the editor can be used by emergency planners for plans specification. During the domain engineering phase, the emergency plan features are chosen, e.g. emergency resolution procedures for team coordination and business process management systems for editor implementation and procedures enactment. The customized emergency plan editor is obtained through the application engineering phase by exploiting the variability model.

Although DPL(EP) supports variability through feature modeling, still has open issues regarding the selection and classification of the appropriate features. These are error prone tasks, since they depend on tacit knowledge require a more refined and consistent formalization process (Johansen et al, 2010; Czarnecki, et al, 2006). To overcome these shortcomings of the DPL method, we propose the usage of well-founded ontologies for feature modeling generation.

WELL FOUNDED DOMAIN ONTOLOGIES

A domain ontology is called well-founded if it is based on a foundational ontology. This means the ontological primitives used in the elaboration of the domain ontology are well-founded and therefore are deemed adequate to properly capture and formalize some fraction of concepts used by a community to communicate about a domain of interest. As a special type of conceptual model, it is intended to be used by humans for purposes of understanding and communication (Mylopoulos, 1992).

Domain ontologies serve a variety of purposes, which include: (i) knowledge-based applications (Guizzardi, 2005); (ii) semantic interoperability (Gonçalves, Guizzardi and Filho, 2011); and (iii) standardization of a shared vocabulary for a community (Nardi et al, 2013); This paper presents a novel application of domain ontologies: for the generation of feature models in the context of document product line for emergency plans.

OntoUML is a language designed to support the creation of well-founded domain ontologies. The language's meta-model complies with the ontological distinctions of a well-grounded theory, named the Unified Foundational Ontology (UFO) (Guizzardi, 2005). OntoUML's rich syntax guides users throughout the modeling activity, which fosters domain knowledge discovery. Besides the language, the OntoUML approach offers a software environment for model construction, verbalization, code generation, validation, named OntoUML Lightweight Editor (OLED1). OntoUML has been successfully adopted in many domains, such as News Information Management (Carolo and Burlamaqui, 2011) and Bioinformatics (Gonçalves, Guizzardi and Filho, 2011).

UFO makes a fundamental distinction between **Particular** and **Universal**. The former is a category that corresponds to individual things, whilst the latter stands for patterns of features multiple individuals have. We say that a Particular (e.g. "John", ISCRAM'15), is an instance of a Universal (e.g. "Person", "Conference"). Particulars are further categorized as **Endurants** and **Events.** Endurants are "wholly present" whenever they are present, i.e., they are in time, (e.g., a house, a person). Events, in contrast, are particulars composed of temporal parts, i.e., they happen in time, in the sense that they extend in time accumulating temporal parts (e.g., a football match, an earthquake) (Guizzardi et al, 2013). Events are possible transformations from a portion of reality to another, i.e., they may change reality by changing the state of affairs from one situation to another.

The Endurant category is further refined into **Substantial** and **Moment**. Substantials are existentially independent Particulars, whilst Moments are

[1] Available at: https://code.google.com/p/ontouml-lightweight-editor/

Particulars that can only exist in other Particulars. Examples of Substantials include "John" and his house; examples of moments include a marriage and the color of an object. Moments are existentially dependent on other individuals. **Modes** are moments dependent on a single individual (e.g. an intention) whilst **Relators** are dependent on a plurality of individuals (e.g., an agreement).

All substantials follow an identity principle, a sort of function that allow us to count individuals and differentiate them. A **Sortal** is a type of Substantial whose instances all follow the same identity principle (e.g. person, rock), whilst a **Mixin** characterizes individuals that follow different principles (e.g., client, agent). **Kinds** characterizes essential properties of Particulars, while **Roles** characterizes contingent properties and are relationally dependent, i.e., its instantiation is determined by a relational property of the individual. The Mixins are specialized into **Category**, those that generalizes Kinds and other Categories and **RoleMixin** that generalizes Roles from different Kinds.

Lastly, **Higher Order Universal** (hou) are universals whose instances are also universals. Examples of higher-order universals are “Dog Type” (whose instances could be “Pointers” and “Retrievers”), and “Installation Type” (whose instances could be “Nuclear” and “Farm”).

EMERGENCY PLANS SUPPORTED BY WELL-FOUNDED ONTOLOGY

This section introduces, at first, the main parts of the emergency plan ontology (OntoEmergePlan). Then, we discuss how it can be applied in the DPL(EP) methodology.

OntoEmergePlan: a well-founded emergency plan ontology

OntoEmergePlan is the main fragment of OntoEmerge, a well-founded domain ontology developed considering the analysis of emergency management processes from England, Australia, USA, etc. It also considered the reuse of elements found in the work of Bringuente et al (2011) and Kollarits & Wergles (2006). Ontological engineering was applied to achieve a higher level expressivity and consistency, considering OntoUML. The main goal of OntoEmergePlan is to support the generation of emergency plans, serving as a reference for common understanding and communication among people, systems and organizations. It brings definitions to respond competency questions such as what is an emergency, emergency types, processes, plans, plans compositions, installations and evacuation routes, involved business activities, types of risks and hazards, among others. The main concepts required for plans generation are illustrated in Figure 2 (some concepts were hidden). One can notice the stereotypes representing UFO categories on the top of each concept. Names above the stereotype are the supertypes of the class in the box. For example, *OE Process* is classified as **Mode**, a special type of intention, where “OE” term means “Organization Emergency” and “E” is “Emergency”. To facilitate reading, the ontology concepts are written here in *italic* and in Figure 2 in **bold**.
*Organization*s are responsible for protecting their workers and their property from hazards. Thus, a self-protection process must be established by defining a set of activities needed to address the hazards. The emergency plan, condensed into a self-protection plan, an *OE Process Definition Document*, is essential to guide to the necessary protection of workers. The plans may include identification of hazards, responsible, resources to carry these actions, among others. The concept of *Installation*, i.e. establishment, corresponds to the total area under control of a *Holder* which develops a *Business Activity*.

An *Environment* (forest, city, farm) is exposed to *Hazardous Event* that can cause *Impact* (destruction, pollution, death). The *Environment* is composed of *Component* (river, building) and an *Installation* is a *Physical Component*. The *Holder* can be a *Person* and/or *Organization*, playing the role of *Businessman*, i.e. an entrepreneur (develops a *Business Activity*), or *Owner* (has *Possession*) of an *Installation*. The same *Holder* may play both roles simultaneously, illustrated by the annotation “{overlapping, complete}”. An *Installation* must have a deployed self-protection plan.

Regulatory Organization, typically, recognizes the *Emergency Standard Process Definition Document* (*ESPDD*) with minimum content that self-protection plans should contemplate. Thus, standard self-protection process guides the preparation of plans according to *Business Activities Type* and *Installation Type*. The former classifies *Business Activity* and the later classifies *Installation*. For example, the NBA regulation is a standard document, i.e., a template of self-protection plan,

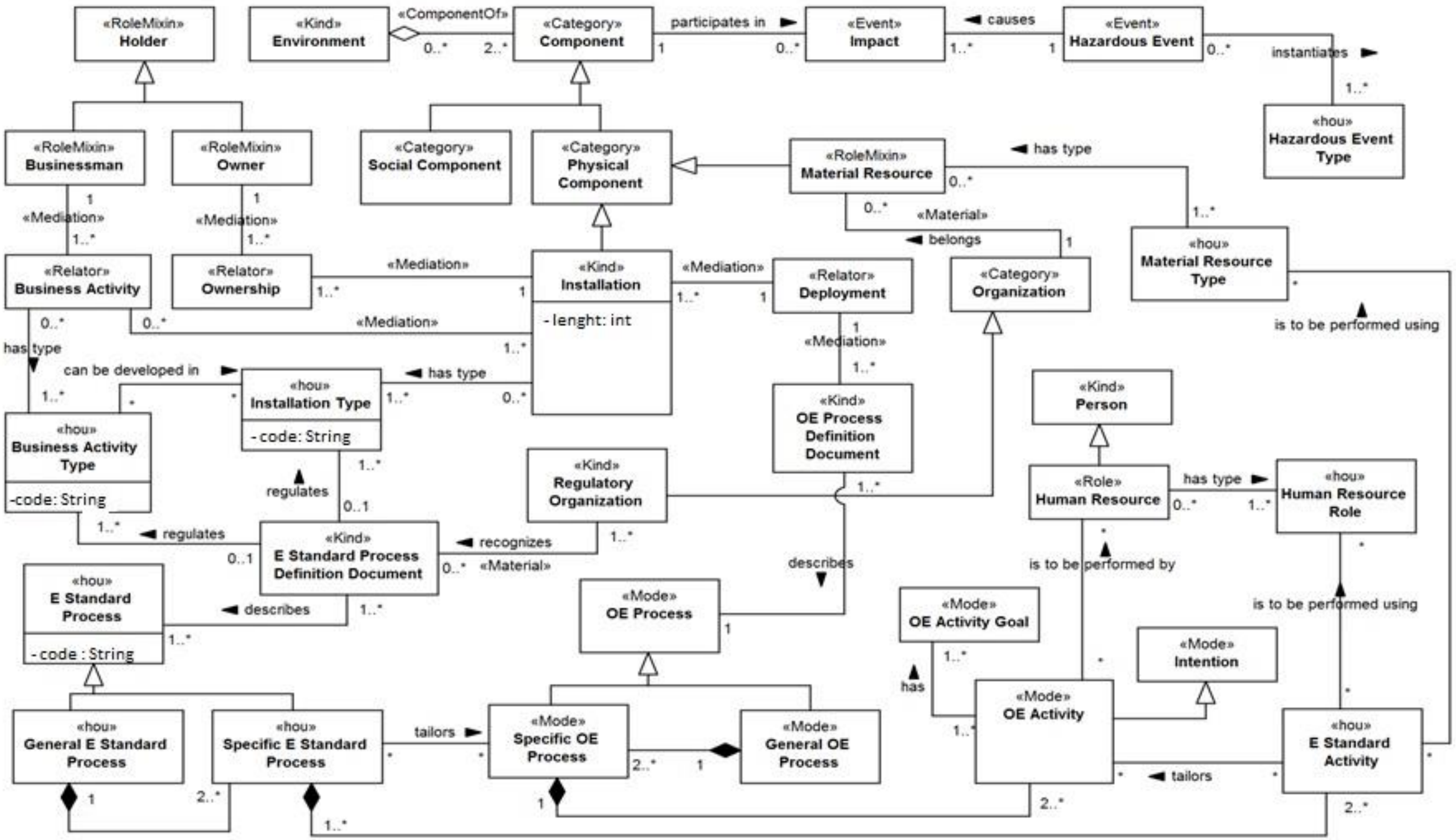

Figure 2. OntoEmergePlan: a well-founded emergency plan ontology

containing the basic elements that must be common to all self-protection plans. An *ESPDD* is a standard self-protection plan that describes the *E Standard Process* (self-protection process). This process can be a *General E Standard Process* or *Specific E Standard Process*. Each *Specific E Standard Process* consists of two or more *E Standard Activity*. The activities are classified into *Complex E Standard Activity*, when the activity is composed of more than one activity, or *Atomic E Standard Activity*.

An *ESPDD* defines a set of rules recognized by at least one *Regulatory Organization* and regulates the *Business Activity Type* and the *Installation Type.* Each *Organization* can tailor, i.e. adapt, the standard process definition to its particular conditions. In this case, an *OE Process* is created and is materialized in the form of an *OE Process Definition Document (OEPDD).* Similar to the *E Standard Process*, *OE Process* may be *General OE Process* or *OE Specific Process*, depending on the scope and composition. The *OE Process* can be composed by one or more *OE Specific Process* and can be decomposed into *OE Activities*.

An *OE Activity* requires *Human* and *Material Resources* to be performed. *Human Resource* is the role that human agents play during an activity, such as fireman and policeman. *Material Resource* includes any equipment required to perform an activity, such as an ambulance. An *E Standard Activity* is performed by one or more *Human Resources*. Likewise, *E Standard Activity* is carried out using *Material Resource Types*, for example the A1 extinguisher, a type of fire extinguisher. Some of these rules are formalized in the ontology through relations (cardinalities) and others as OCL constraints.

DPL(EP) supported by OntoEmergePlan

We propose the use of OntoEmergePlan in the DPL(EP) methodology to enhance feature modelling when analysing a document family, during the domain engineering phase. In this phase, the domain ontology should cover the similarities and variability of product line (Mohan and Balasubramaniam, 2003) and can be mapped into a feature model. (Czarnecki et al, 2006).Thus, the domain knowledge, consistently represented in an ontology, can provide the basis to derive rich rules associated with features model. In addition, we believe that the well-founded domain ontology can facilitate the identification of the similarity and variability points of the document family by incorporating the ontology meta-characteristics (e.g. dependencies, cardinalities, OCL rules).

The feature model provides an overview of the requirements and represents the features of documents and their dependencies (can be understood as a tree with alternatives and variation points as its nodes), the dependency type (mandatory or optional), the selection type ('or' or 'exclusive-or') and restrictions 'required' or 'exclusive'. The nodes represent valid configurations (i.e. a set of values assigned to a set of variation points).

The interpretation of a class model as a feature model depends on mapping rules. Some thoughts about these mappings are cited in Johansen et al. (2010). An example of a semi-automatic tool that implements them is the graphic editor for feature modelling of Gómez and Ramos (2010). It gives an idea how this process might work, allowing rules specification through OCL constraints being compatible to the Domain Variability Model (DVM) format. Developers can define a document product line generating editors that specify feature model configurations.

Figure 3 illustrates the self-protection plan generation by using the DPL(EP) adapted in our proposal. The first step indicates the generation process of the emergency feature model supported by OntoEmergePlan. The structure of the feature model is defined by mapping rules and constraints as OCL expressions to the emergency feature model. For example, the restriction: “if the business activity type is ‘transportation infrastructure’ and installation type is ‘railway tunnel’ and the installation length is greater than 1000 meters, then the standard process code should be ‘RD393’ or ‘RD635’” is formalized as an OCL constraint. They are inserted by annotations and used in the next steps of the prototype to automatically verify if instances of the feature model are valid. The mapping can start by specifying the feature root as the product line name (self-protection plan). The feature leaves are added according to class structure and their relationships. In this initial experimentation, some variation points (nodes) could be identified in the feature model:

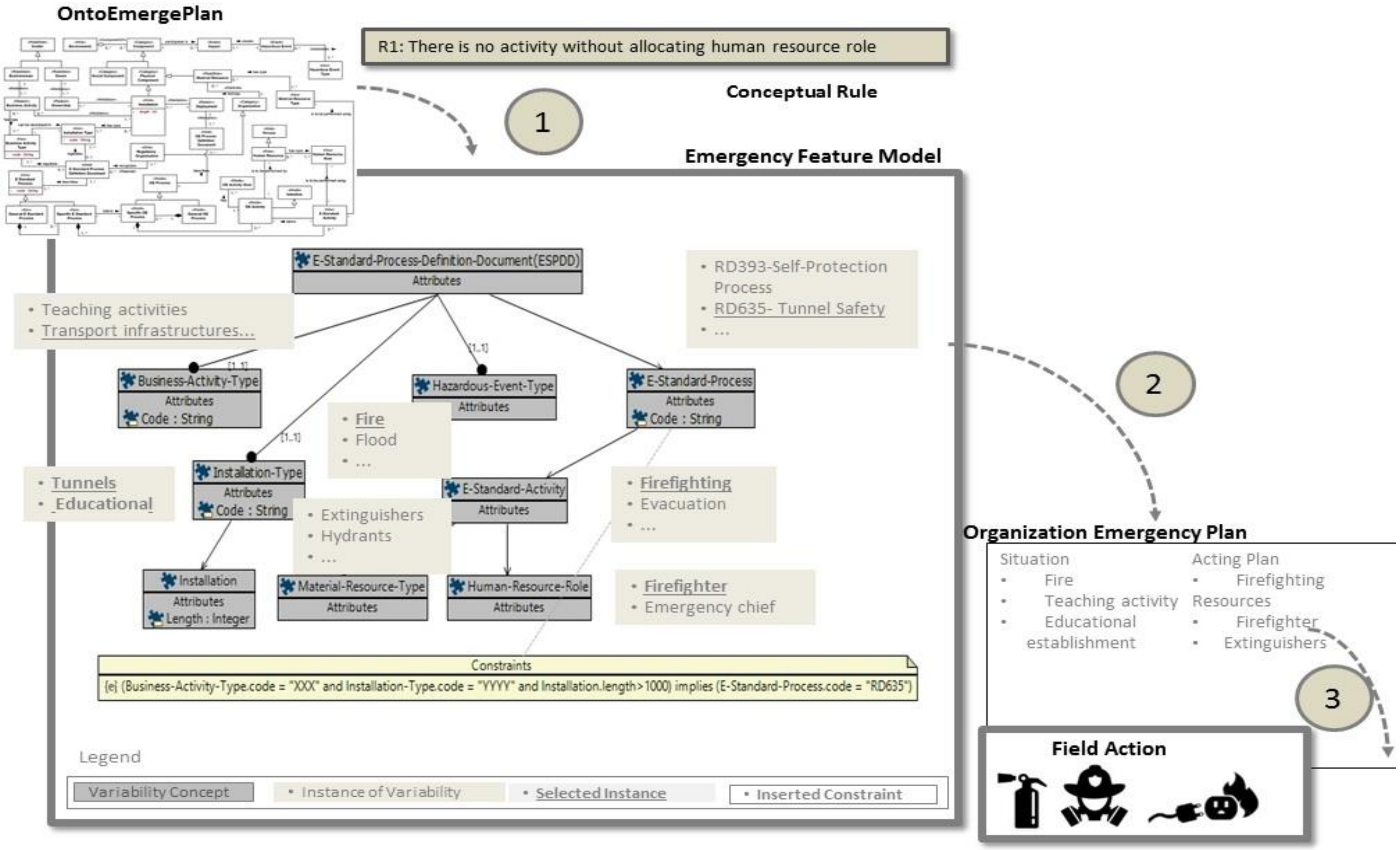

Figure 3. DPL(EP) Method based on OntoEmergePlan ontology

(a) the Business Activity Type, obtaining the corresponding regulation and the standard document with the definition of the contents of the emergency plan. The type of business activity also defines the associated risks;

(b) Installation Type, which defines Installation composition, construction characteristics and related Risks;

(c) Emergency Event Type, which defines the operation procedures to be performed and resources to be used.

The plan generation supported by this tool occurs in the second step, where the OE Plan is produced. Finally, the third step illustrates the plan being used during the emergency response.

CONCLUSION

In this paper we introduced an approach to enhance the emergency plans quality and its semantics by adapting the methodology DPL(EP). To address semantic problems, ontologies have been previously applied in feature modeling. However, these ontological approaches didn't focus on the capture of general perception, which is commonly addressed by well-founded domain ontologies. Therefore, we propose the application of a well-founded ontology during the feature modeling of the emergency variable aspects. A well-founded domain ontology counts on ontological assumptions from a foundational ontology and the feature model can incorporate its benefits, increasing the semantic expressiveness of the design activity and the quality of the feature model itself. Consequently, we believe that a more consistent plan generation can be achieved with our approach.

A well-founded ontology, OntoEmergePlan, was presented. It defines concepts such as hazardous events, emergency processes and activities, material and human resources, regulatory organizations, installations, among others. DPL(EP) was adapted to use of the ontology, which provides a precise conceptualization and vocabulary of the domain, supporting the designer during the feature modeling. A common understanding formalization brings consistency to the final feature model, enhanced by the constraints coming from the ontology as conceptual rules defined as OCL statements.

The contributions of this work can be summarized as: (i) a proposal to adapt the existent method DPL(EP) to increase emergency plans quality and; (ii) the well-founded emergency ontology OntoEmergePlan, which provides enough clarity to a community of users for better communication, also supporting systems development.

As future work we intend to evolve OntoEmergePlan ontology and apply it in other examples of emergency plans generation through the DPL(EP) method. In addition, we believe that the integration of this approach in a feature modeling tool following model-driven engineering can automate the whole process, reducing human failures during emergency plans documentation. In the prototype, the feature model derivation from the domain ontology is now being automated, .Mapping rules to support this can be found at Johansen et al. (2010).

REFERENCES

1. Boin, A., Hart, P. (2010). Organising for Effective Emergency. Management: Lessons from Research. *Australian Journal of Public Administration*, 69,4, 357–371.
2. Bringuente, A.C.O., Falbo, R. A. and Guizzardi, G. (2011) Using a Foundational Ontology for Reengineering a Software Process Ontology. *Journal of Information and Data Management*, 2, 511-526.
3. Czarnecki, K., Peter Kim, C. H, and Kalleberg, K. T. (2006) Feature models are views on ontologies. *SPLC '06. IEEE*, 41-51.
4. Gómez, A. and Ramos, I. (2010) Cardinality-based feature modeling and model-driven engineering: Fitting them together. *Proceedings of the Fourth International Workshop on Variability Modeling of Software intensive Systems (VAMOS'10)*, Linz, Austria.
5. Gonçalves, B., Guizzardi, G. and Filho, J. G. P. (2011) Using an ECG reference ontology for semantic interoperability of ECG data. *Journal of Biomedical Informatics*, 44, 1, 126–136.

6. Guizzardi, G. (2005). Ontological Foundations for Structural Conceptual Models. PhD thesis, University of Twente, The Netherlands.

7. Guizzardi, G., Wagner, G., Falbo, R. De A., Guizzardi, R. S. S. and Almeida, J. P. A. (2013) Towards Ontological Foundations for the Conceptual Modeling of Events. *Proceedings of the Thirty Second International Conference on Conceptual Modeling (ER)*. Hong Kong, China:

8. Johansen, M.F., Fleurey, F., Acher, M., Collet, P. and Lahire, P. (2010) Exploring the Synergies Between Feature Models and Ontologies. *Proceedings of Forteenth International Workshop on Model-driven Approaches (MAPLE) in Software Product Line Engineering (SPLC'10)*, Jeju Island, South Korea.

9. Kollarits, S. and Wergles, N. (2006) Monitor Ontology Report - an ontological basis for risk management - Risk Ontology - INTERREG IIIB Cadses project supported by the European Regional Development Fund (ERDF). Retrieved January 30, 2015, from: http://www.monitor-cadses.org/documents/MONITOR_BaseOntology_Report_1_0.pdf

15.

10. Mylopoulos, J. (1992) Conceptual Modeling and Tellos. In: Loucopoulos, P.; Zicari, R. (Eds.). *Conceptual modeling, databases, and case: An integrated view of information systems development*. New York: John Wiley & Sons.

11. Mohan, K. and Balasubramaniam, R. (2003) "Ontology-based Support for Variability Management in Product and Service Families". *Proceedings of Thirty Sixth Hawaii International Conference on System Sciences*, Hawaii.

12. Nardi, J. C., Falbo, R. de A., Almeida, J. P. A., Guizzardi, G., Pires, L. F., Van Sinderen, M. and Guarino, N. (2013) Towards a Commitment-Based Reference Ontology for Services. *Enterprise Distributed Object Computing Conference (EDOC).* Vancouver, Canada: IEEE , 175–184.

13. NBA (2007). Retrieved February 26, 2015, from: http://www.boe.es/boe/dias/2007/03/24/pdfs/A12841-12850.pdf

14. 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. *Proceedings of the 8th International ISCRAM Conference*, Lisbon, Portugal, 1-10.