University of Wollongong

Research Online

Faculty of Engineering and Information
Sciences - Papers: Part A

Faculty of Engineering and Information
Sciences

1-1-2013

Model-driven disaster management

Siti Hajar Othman

Universiti Teknologi Malaysia, sho492@uow.edu.au

Ghassan Beydoun

University of Wollongong, beydoun@uow.edu.au

Follow this and additional works at: <https://ro.uow.edu.au/eispapers>

Part of the [Engineering Commons](#), and the [Science and Technology Studies Commons](#)

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au

Model-driven disaster management

Abstract

Disaster management (DM) is a challenging domain to model because of the variety of dynamic characteristics attached to the domain. Metamodeling is a model-driven approach that describes how semantic domain models can be built into an artifact called a Metamodel. By collecting all the domain concepts and partitioning the domain problems into sub-domain-problems, a metamodel can produce a domain-specific language. This paper presents a Disaster Management Metamodel that can serve as a representational layer of DM expertise. This metamodel leads to better knowledge sharing and facilitates combining and matching different DM activities to best manage the disaster on hand.

Keywords

era2015

Disciplines

Engineering | Science and Technology Studies

Publication Details

Othman, S. Hajar. & Beydoun, G. (2013). Model-driven disaster management. *Information and Management*, 50 (5), 218-228.

Model-driven DM

Abstract

Disaster Management (DM) is a challenging domain to model because of variety dynamic characteristics attached to the domain. Metamodelling is a model-driven approach that describes how semantic domain models can be built into an artifact called a Metamodel. By collecting all domain concepts and partitioning the domain problems into sub-domain-problems, a metamodel can produce a domain specific language. This paper presents a Disaster Management Metamodel, the metamodel that can serve as a representational layer of DM expertise leading to better knowledge sharing that can also facilitate combining and matching different DM activities to best manage the disaster on hand.

Keywords – Model-driven approach, Disaster Management, Metamodel, Model, Metamodelling, Knowledge management

1.0 Introduction

Disaster management (DM) is the management of risks and consequences of a disaster. This includes various aspects of planning and responses in all phases of a disaster event: mitigation, preparedness, response and recovery (W3C Incubator Group, 2008). DM is a knowledge *domain* which can also be defined as the effective organization, direction and utilization of available counter-disaster resources (Asian Disaster Preparedness Center (ADPC), 2002). The aims of this domain practitioner are to reduce and avoid potential losses from hazards, assure prompt and appropriate assistance to victims of disasters and achieve a rapid and effective recovery. Many complex series of activities co-exist in the operationalisation of this knowledge domain. These activities include risk assessments, preparedness actions, emergency responses, rescue operations, aid distribution, reconstruction tasks and many others. The DM knowledge domain is indeed quite complex, both to model and communicate and moreover it is continuously evolving. Rather than aim for a comprehensive and a complete model, in this study we propose a metamodel which can pull together the various, disparate and partial models that attempt to describe the DM knowledge systematically.

Models have been used in many areas to share and communicate knowledge about the world (Jeusfeld, Jarke, & Mylopoulos, 2009) or manage complexities (Levendovszky, Rumpe, Schatz, & Sprinkle, 2010). They can structure the theory of generic concepts that shape the way we conceive our phenomenon in our reality

(Hommes, 2005). Ideally, models have a causal connection to the modeled part of reality (Abmann, Zschaler, & Wagner, 2006). They must form true or faithful representations so that queries of the model give reliable statements about reality or manipulations of the model result in reliable adaptations of reality. A *model-driven* approach can offer many advantages of modeling details at varying levels of abstraction and allows integrated flow of information. In this context, metamodeling is a technique used in the software development field which describes how a model can be built (Nordstrom, 1999). A metamodel is the artifact output of metamodeling that makes statements about what can be expressed in the valid models of the knowledge domain (Seidewitz, 2003). It is often defined as a model about models. In our context, a model refers to the *DM solution model* which can show the coordination of DM activity and its elements (e.g.: human, resources, plans) and how these should be arranged for a specific disaster.

A metamodel can identify specific domain features, collecting all domain concepts and partitioning the domain problems into sub-domain-problems. This can help zoom in and identify sources of inadequate DM practices. Failures in DM are in fact often due to an accumulation of complex chain of events and often accompanied by changes in external environment factors (Aini, Fakhrol-Razi, Daud, Adam, & Kadir, 2005). The management process itself is also contingent on existing organisational structures charged with the DM processes. It is common wisdom that no two disasters are exactly the same, and that every disaster requires its own management process. However, the way disasters impact human lives and business processes exhibit similarities and responses are often transferrable between disasters. Evacuation of personnel for example is a DM action that is applicable in many disaster situations. For this research, we structure the complexity of DM activities by developing a metamodel called a Disaster Management Metamodel (DM Metamodel). It is a precise definition of the constructs and rules needed for creating DM models. It will be a representation of how all DM models can be constructed (e.g.: a mitigation model of bushfire, a response model of earthquake disasters, an aid distribution in nuclear disasters and many more). To develop the metamodel, we identify all generic concepts that appear in DM domain (e.g.: emergency team, rescue resource, emergency plan, Evacuation, Disaster monitoring and alert warning). The aim of the DM Metamodel is to allow its users to easily create specific DM solution models based on their own disaster challenges. This will not only support users making quicker decisions, but it will also provide DM knowledge sharing among varying DM communities. The research presented uncovers the potential of the model-driven approach to support DM. For demonstrating its applicability, we use two recent and real world disaster scenarios: the 2011 Christchurch Earthquake disaster in New Zealand and the recent Nuclear Meltdown in the aftermath of the 2011 tsunami in Japan.

DM requires *decision-making* activities in the operation of its domain. Weighing the amount of information needed before making a decision against the time available is a challenge. Timely decision-making to direct and coordinate the activities of other people is important in order to achieve the DM goals. Making rapid decisions in the

chaos during disaster events is a very challenging endeavor. The process can become increasingly difficult when a decision includes considering the specific interests of victims, governments, NGOs and other emergency services teams. We believe if all DM processes, tasks and coordination are comprehensively and explicitly formulated, the potential to resolve many DM decision makers' problems is improved. DM knowledge domain is also dispersed. As a metamodel can describe other models, it can provide a clear representation of how various DM solution models can be generated. The metamodel must not only provide the solutions for different disaster activities, but it must also be able to handle different types of disasters. A solution model is expected to describe roles and functions that need to be performed by the DM users in their specific scenarios. This approach can help many DM users e.g.: Emergency Managers, Monitoring users, Local and state government, Emergency Service Teams, Aid Agencies.

The rest of the paper is structured as follows: The first part describes related work to further highlight how the use of metamodel can facilitate managing problems in DM. The second part briefly describes the development process of the DM Metamodel. The third part presents the process to assess the conformance of real world DM solution models to our DM Metamodel. The fourth part presents the implementation of the conformance process in two disaster scenarios: (i) the Earthquake disaster in Christchurch, New Zealand and the Nuclear Meltdown in Fukushima, Japan. Finally, the last part concludes the paper with a discussion on our findings and future works of this research.

2.0 Related Work

“Structuring information and maintaining it, takes time and effort and requires a type of quality control (Heghe, 2011) pp.38 ”

The use of metamodeling in software development creates interoperable, reusable, portable software activities and components. Adapting the success of this approach into DM can offer similar benefits. Our work is also inspired by the organization of *software factory* where a software product line will use a special software factory *template* based on an existing schema (Greenfield, Short, Cook, Kent, & Crupi, 2004). The template is used to configure extensible tools, processes and content to automate the development and maintenance of variants archetypical product. This process can be implemented by adapting, assembling, and configuring framework-based components. We adapt this idea into the organization of DM domain knowledge. Generally, by developing a metamodel for a specific domain, therefore, language of the domain can be modeled efficiently. To develop a language for a DM domain is of the objectives that this research tries to achieve. Metamodel has a capability to *check the completeness* of the domain it models. According to (Lalonde, 2011), the author developed a conceptual framework of crisis management that could help in

strengthening the resilient capabilities of individuals, organizations and communities to face a crisis. However, the theoretical framework developed in that study needs to be empirically validated by some experts in the field. In such a scenario, we believe a DM metamodel has a capability to validate the completeness of all concepts used in the Lalonde model. This is possible because the metamodel generalizes all the concepts that appear in DM models during its development. We believe the metamodel we develop has the potential to be used as a standard guideline for DM knowledge modelling.

To develop a good software-based support system, a collection of elements which are organized together for the purpose of developing the system is important. As defined by (Alhir, 2003), *model of a system* is a description or specification of the system and its environment for a specific purpose. The most important task of developing this specification is to gather all concepts that are important for the purpose of organizing the system. In our context, the specification as mentioned by (Alhir, 2003) is the metamodel that we developed for DM. Developing the metamodel allows the DM users to derive the best disaster solution model from the specifications provided. Understanding, designing, constructing, deploying and modifying models to best describe the system is the main focus of the model-driven approach. The same situation can be applied to the DM domain if a good DM organization system is to be achieved. For example, if one country wants to create a good flood evacuation process, all DM components on how the process needs to be executed must be presented clearly. This can be constructed by a model that can abstract the real evacuation processes. The processes could be a combination of activities on how: (i) the people at risk should evacuate from their disaster hit location, (ii) the emergency services team should coordinate the evacuees, (iii) to setup the evacuation operation centers, (iv) to organize the evacuation centers and other evacuation processes.

Figure 1 Problem Solving through Model-Driven approach (extended from (Alhir, 2003)).

To find the best solution model for various disaster problems is not an easy task. The process is akin to executing a *Problem Solving* process. This requires sub-processes for understanding problems, solving problems and implementing the solutions (Alhir, 2003). Figure 1 shows how a model-driven approach through a metamodel corresponds to a problem solving environment. By developing a metamodel specific for the DM, we uncover and make explicit key aspects of activities, organization, resources and all elements in DM. These are the environment components of the domain (e.g. users, resources, procedures or plans) which need to be precisely identified by the metamodel. According to (Beydoun et al., 2009), a unified metamodel can ensure that the key concepts of a domain are easily presented to newcomers to the domain. A metamodel can also create a better communication amongst practitioners and researchers about the domain. This can lead to improving and realizing of a unified body of knowledge of the domain.

Our work supports DM users with structured constructs and rules of DM knowledge (e.g.: concepts, activities, stakeholders or resources). In (Alexander, 2002), the author agreed that to better organize DM coordination, users in DM need to really understand the methods, procedures, protocols and strategies of the emergency planning. The consequences from the organization of DM knowledge and DM Metamodel can offer many benefits. Some of these advantages are: (a) to facilitate global communication among different disaster emergency users as the metamodel has generalized all the concepts that must exist in DM domain; (b) provide guidelines for creating comprehensive DM models (e.g: *Flood Emergency Response Model* or *Tsunami Risk Reduction Model*); (c) Enable users to create new customized DM models based on choosing and combining a set of component concepts based on their disaster problems; (d) Simplifying instructing new created solution models among DM users because a set of syntax and semantic rules of the domain is provided and (e) Highlight the scope for improvement in DM practice through the validation of metamodels against other DM models. Consequently, a model-driven approach through a metamodel could help DM users to understand DM operations more easily.

Three important elements are required for the realization of a model-driven approach: a *model*, a *metamodel* and *model transformations* (Trabelsi, Atitallah, Meftali, Dekeyser, & Jemai, 2011). A *model* is the abstract representation of domain in the real world and has two key elements: *concepts* (characterizes *things*) and *relationships* (characterizes *links* between these things in the real world). A *metamodel* is a collection of concepts and their relationships that describe the models. It uses a model descriptive language and defines the syntax of models. The third element, *model transformation*, is the process of transforming the elements of a source model that conforms to a particular metamodel into elements of another model (target model) that further conforms to a metamodel (OMG, 2002b). Through metamodeling, the model-driven approach allows a modeling of domain models that can be performed at different levels of abstraction. In this research, we apply the Meta Object Facility (MOF) metamodeling framework (OMG, 2002a).

Among the key reasons to build a metamodel are to understand why complex model behaves as it does and to explore the behavior of the model against a large part of its domain (Davis & Bigelow, 2002). For the development of DM metamodel, we realized that the *biggest challenge* of this task is to identify the general concepts used in all phases of the DM. By identifying the DM concepts, the metamodel to be developed can partition all the DM problems into sub-DM problems. After the concept is identified, all elements hidden behind the concept (e.g.: who performs the concept, how the concept should be performed, when it should be performed, what are the pre-requisites before performing the concept) must be obtained. The concept then is reconciled to ensure its generality. Particularly for DM, the domain contains a variety of elements in each of its domain operation. These elements include DM tasks, DM activities, DM roles, DM resources, DM decisions, DM users, DM tools or even the unpredicted DM environmental events. (Varro, 2004) agreed that a model-driven can enable many organizations to integrate whatever they already have in place (disaster solutions) with whatever they build today (current problem) and whatever they will build for tomorrow (disaster solution cases for future use). For instance, in a software development field, the model-driven approach is applied by using a specific software language to construct complete software models. The language helps many software developers to develop a variety of software models correctly. This is also possible for the DM domain if the construction of its language exists.

Figure 2 Domain, model and metamodel elements. (Stahl, Voelter, & Czarnecki, 2005)

Relationships between model, metamodel and real world domain is illustrated in Figure 2. To better understand the relationship among these elements, the relationship between computer program and its programming grammar is an illustrative analogy: The task of a computer programmer (DM user) is to program a computer program (model) with the correct programming grammar (metamodel). Only then can the computer program be executed correctly. The developed computer program is said to conform to its programming grammar. Similarly, when DM models are developed conforming to a DM Metamodel, a creation of correct DM models can be ensured. This eventually will bring to the realization of good DM organization models in their real world domain application. In our earlier work, (Othman & Beydoun, 2010b) presented the motivation of adapting metamodeling for DM domain. That work was continued by performing an iterative metamodeling process to the DM domain as appeared in (Othman & Beydoun, 2010a). The result from that metamodeling process is the DM Metamodel that we present and illustrate its use in actual disaster case studies in this paper.

3.0 Metamodel for the Disaster Management Domain

In this section, an overview of the DM Metamodel development process and the resulting metamodel are discussed. Later, the focus will be on the actual usage of the DM metamodel. It is developed based on the *Four-layer* metamodeling framework of the Meta Object Facility (MOF) offered by the Object Management Group (OMG, 2002a). In this framework, the four layers are labeled as M3, M2, M1 and M0-level. The details of each layer are as follows:

- (i) M3-level is reserved for *Meta-metamodel* elements modeling concepts;
- (ii) M2-level is a reserved for *Metamodel* elements (instance of *meta-metamodel*) which define a language for specifying models.
- (iii) M1-level is a layer for *Model* elements (instance of *metamodel*) which define a language that describes an information domain and
- (iv) the lowest level, M0 is specifically dedicated for *User Model* elements (instance of *model*) which describe a specific information domain.

The MOF framework can offer many advantages to the users of the metamodel. As described by (M. Picka, 2004), it can do the following: (a) support any kind of model and modeling paradigm imaginable, (b) allow different kinds of metadata to be related, (c) allow metamodels and new kinds of metadata to be added incrementally and (d) support the interchange of arbitrary models and metamodels between parties that use the same *meta-metamodel* (M3-level). Specifically for a metamodel positioned in the M2-level, the work of (Davis & Bigelow, 2002) discusses six criteria to gauge the *goodness* of a developed metamodel. These criteria include: (i) prediction other metamodel reasonably consistent with those of the baseline model across the domain, (ii) having independent meaningful variables, (iii) ability to identify/highlight all input variables that are essential to success (critical components of a domain) (iv) reasonable depiction of relative ‘importance’ or a generate of statistical measures of the significance or importance of candidature variables and (v) provision of a good storyline (could tell the users why and how the model should behave as it does).

To create the DM Metamodel, a metamodeling creation process from the work used to develop a *Framework for Agent Modelling Language* (FAML) in (Beydoun, Low, Mouraditis, & Henderson-Sellers, 2009) and (Beydoun, Low, Henderson-Sellers, et al., 2009) was adapted. Some of the steps we took for the DM Metamodel development are: (i) extracting the general concepts relevant to all identified DM Models; (ii) short-listing the candidate concepts; (iii) reconcile the differences between concepts and (iv) identify the relationships among the concepts. Before those four steps are executed, a preliminary step is first taken to identify DM models which are deemed influential. The extent of the influence of a given model is estimated using a heuristic function that combines the model acceptance (as indicated by citations to the model) and the effort that has gone in developing it (the size of the organisation behind it). The heuristic function was recently developed in (Othman & Beydoun, 2010a). The function normalizes the impact of the model to its longevity. Once a model is selected, disaster specific expertise is deleted and only general DM concepts are shortlisted for

In Figure 3, we present the resultant of DM Metamodel as the output after all steps to create it is applied into the metamodel.

Figure 3 The DM Metamodel

The DM Metamodel contains the relationships among concepts which can represent the semantic of DM domain as shown in Figure 3. The core class in this DM Metamodel is the *Organisation* which represents the loose ‘organization’ where DM concepts are operationalized. All key concepts in DM are grouped in the *Organisation* concept. Other key DM concepts are aggregated within this class and they include: *DMPcedure*, *DMRequirement*, *DMPolicy*, *Actor*, *ElementsAtRisk*, *DMTeam*, *DomainKnowledge*, *Resource*, *ActorRole* and *MessageCommunication*. *DMPcedure* can represent the collections of implemented procedures of DM activities including for example Mitigation, Preparedness, Rescue, Response and Evacuation. *DMTeam* defines a collection of *ActorRole* class which typically describes human roles that work towards a *DMGoal*. *ActorTask* class in our metamodel is derived from a *DMGoal* class. Here we also model a *DisasterPreventionGoal* as a class that can be achieved by *DisasterPreventionTask*. *ElementsAtRisk* includes elements that bear the brunt of a

disaster impact. These include infrastructure, natural sites and people. Of these elements at risk, people can be active elements that also can play a role in the DM process. Specifically, the ‘victim’ victims can be active elements in DM and can have direct impact on the process of DM. This impact can be positive as well as negative on the effectiveness of the DM process. This is a dynamic relation between victims and the unfolding of the DM process and is a recurring aspect of DM that it warrants an explicit inclusion in the metamodel. As such, ‘*Victim*’ is an important subclass of ‘*People*’ at risk associated with both concepts, disaster ‘*Indicators*’ and ‘*Actor Task*’ (components of the DM goals).

After the metamodel is developed, the next stage is to determine how the metamodel can be used by the users of the metamodel. In our context, the DM Metamodel has a potential to instantiate a new DM model based on the problem requirements given by the DM users. In the next section, we detail how this process can be realized by showing its applicability in the real world of DM.

3.1 The Conformance of DM Model from its DM Metamodel

In the MOF metamodeling framework, the derivation of a model from one level to another level is a ‘model transformation’. In (Henderson-Sellers, 2011), the transformation of model from its metamodel is called a ‘*conformance*’. Through the conformance process, a realization of concept in DM Metamodel to be its new instance (object) in Model at M1-level can be achieved. In M0-level, all objects created by a *Model* in M1-level can be used as an *instance* in *User Model* (M0-level). The *instance* can then be stored in a DM Knowledge Repository for the purpose of future use. An instance created in M0-level represents a sample of DM solution model obtained from the Model. The solution model is valuable because the artifact can be shared and be referred by other DM users at other times. It is an important model especially to the DM users who may be dealing with the same kind of disaster problem. Besides, time taken to solve the recurrent problem can be reduced as the same problem has been previously solved by other DM users.

The DM Metamodel is positioned in M2-level in the MOF metamodeling framework. Therefore, a *Model* which is positioned in M1-level can be modeled by the metamodel. All the Models (M1-level) are created based on model specification language described in its conformance *Metamodel* (M2-level). The same process form can also be inherited by the *User Model* (M0) which is positioned in a level lower than the M1-level. In MOF, the domain concept used in a metamodel is presented as a *Class*. The data for a *Class* is presented as an *Object*. And, data for the *Object* is in turn presented as an *Instance* in *User Model*. The *User Model* in M0-level is the *target model* that we aimed from the DM Metamodel. The derived target model presents the model of real-world DM scenarios and it contains ‘human and universal’ elements. Figure 4 graphically illustrates these inter-level relationships.

During the metamodel creation process, we collected and reconciled all DM concepts appearing in the DM domain. The identification of all these concepts enables the DM Metamodel to handle the complexity of the domain. For example, a *DisasterActionService* is one of the DM concepts we identified (refer to Figure 3 for the position of the concept). During the metamodel conformance process, concepts in M2-level can be instantiated to be a new object in the DM Model at M1-level. For the *DisasterActionService* for example, the concept is being instantiated to be the new object of *EarthquakeActionService*. This object is the instance of *DisasterActionService* concept that will be used in the Earthquake Emergency Response Model at M1-level. Second example is the instantiation of *EarthquakeProcedure* from the *DMPProcedure* concept.

Figure 4 A *DM Conformance Framework* shows how DM model (M1-level) can be instantiated from its conformant *Metamodel* (in M2-level). It will then become the *User-Model* (M0-level) representing a target model in real world DM application.

The *DisasterActionService* is created with a *(0..*)* (zero or many) relationship to concept *DMPProcedure* in DM Metamodel. The *'relationship statement'* sets a rule which allows the modeler to create more than one object from the *DisasterActionService*. For example, in Figure 4 the *DisasterActionService* is instantiated into new objects of *EvacuateVictims* and *RescueVictimsInRubble*. Next, we inherit the relationship of *Follows* to be a new relationship to tie between the *EvacuateVictims* and the *StateEvacuationProcedure*. And also, *Follows* is a relationship that ties *RescueVictimsInRubble* and the *StateRescueProcedure*. These instantiations provide a meaning that in a real DM scenario, disaster operation services can be implemented in different DM activities (evacuation and rescue). But all these

activities must follow their own specified procedure in order to perform the right DM action (refer “*Follows*” relationship in M2).

Other than its potential to offer solution for the disaster problems faced by the DM users, the DM Metamodel also allows them to create their own DM model based on the construct and rules provided by the metamodel. For example in the preparedness against bushfires, the Department of Education and Early Childhood Development (DEECD) in Victoria, Australia has requested every school (government and non-government) and children's service in the state to review their own emergency management plan (Department of Education and Early Childhood Development (DEECD), 2010). Together with the task, they also need to address any priority maintenance works for that plan. This is the example of a situation where the DM Metamodel could help many DM users (school) through transforming new DM solution model specific to their own problem requirements. The process is possible as a repository built using the metamodel has stored various previous bushfire solution samples created by previous DM users. Table 1 presents a sample of models in each level which can possibly be derived from the DM Metamodel.

Table 1 Sample of possible DM *Models* and DM *User Models* which can be instantiated from the DM Metamodel

Metamodel (M2-level)	Model (M1-level)	User Model / Real World (M0-level)
DM Metamodel	<ul style="list-style-type: none"> Disaster Mitigation model Disaster Risk Management Disaster Resilience model Disaster Preparedness model Evacuation model Early Warning System model Preparedness Training model Emergency Response model Search and Rescue model Emergency Traffic model Situation Awareness model Responder Collaborative model Aid Distribution model Mass Casualty Model Disaster Recovery model 	<ul style="list-style-type: none"> A model of disaster planning frameworks to respond to major tourism disasters in Filipina A model for the resilience of Southern Africa to disasters. A model of earthquake early warning system framework for Asian Countries. A model of spatial information technology in Flood early warning systems in Queensland. A model of multi-agent teams to improve the training of Wildfire incident. A model of multi-agent simulation of Bushfire A model of Flood emergency response simulation using wireless sensor networks model A model of Aid Distribution in Haiti Earthquake A model of evidence-based care of psychological support for disaster victims after Earthquake A model of Disaster Response after Hurricane A model of Chernobyl Nuclear disaster in Ukraine

3.2 Validation: Applying DM Metamodel to describe Real-World Disaster Management

Our metamodel is generic and generalizes various kinds of disaster concepts that can be refined according to the disaster on hand. We anticipate that various concepts in DM, their relationships and attributes, different types of data models could be generated by using the conformance of DM model from the DM Metamodel. To

illustrate and validate this approach, in this section we present two case studies highlighting how a conformance of the DM model from the DM Metamodel is performed. These cases are two disaster scenarios. A natural earthquake disaster (Christchurch Earthquake, New Zealand) and a man-made disaster (Nuclear Crisis in Fukushima, Japan).

The first case study, for showing the conformance of the DM model from the DM Metamodel, is shown by the model in Figure 5. This Earthquake Model is the specific disaster problem which is modeled for a purpose to proof the potential of our metamodel. It illustrates the conformance of the Earthquake Model that has a position at M1-level in MOF metamodeling from the DM Metamodel. For the purpose of representing the real-world earthquake scenario, we then use the real event of the Christchurch earthquake disaster of February 2010 (shown by *User Model* in Figure 6). This *Christchurch Earthquake Model* and this new *User Model* are positioned at M0-level in MOF metamodeling.

3.2.1 Case Study 1: Earthquake (Natural Disaster)

Figure 5 The Earthquake Model (M1-level) as a conformance result from the DM Metamodel (M2-level).

An ‘Object’ is created when a concept is instantiated from the DM Metamodel at M2-level to its new conformance model at M1-level (e.g. Earthquake Model). As we

Figure 6 Christchurch Earthquake Disaster is the sample of *User Model* (M0-level) for one earthquake real scenario. This *User Model* is the model we refine from the Earthquake model (M1-level)

Next, for the *EarthquakeActionService*, the concept represents services which can be provided by rescue teams during the incident. With the aims to drive the disaster

situation to a more stable state, the *EarthquakeEmergencyTeam* concept needs to provide ('Serves' as the concept relationship) their rescue service (*EarthquakeAction-Service*) to all disaster affected elements (*EarthquakeAffectedElement*). The elements include *AffectedResidents*, *AffectedNaturalSite*, *AffectedCivilianSociety* and *AffectedInfrastructure*. Hidden behind each of these concepts/objects/instances are their working details. For example, the *EarthquakeRescueResource* is one of the objects that appears in the Earthquake Model (M1-level). Behind this object are the following details: (a) what are the resources required, (b) how the resources are arranged, (c) what are the functions of each resource and (d) which authority is responsible for each resource, particularly for earthquake disasters. We believe that the DM Metamodel that we developed is useful for the purpose of creating a range of DM models suitable for varying contexts.

Figure 6 presents the *Christchurch Earthquake Model* as the sample of *User Model* (M0-level) created from the Earthquake Model (M1-level). The model describes the scenario of the recent earthquake disaster which happened in Christchurch, New Zealand in February 2010. Through the model, Object used in the Earthquake Model becomes a new Instance in the Christchurch Earthquake Model. An 'Instance' is represented when the Object is instantiated from the Model at M1-level to its new *User Model* at M0-level (e.g.: Christchurch Earthquake Model). Instances represent Objects in the real world. For example, in the Christchurch Earthquake model the causes of the Christchurch tragedy come from a combination of few factors (as represented by the *EarthquakeGravityFactor* and *Earthquake-ComplexityFactor*). These include (i) the movement of the Australia and Pacific tectonic plates, (ii) the identification of high pre-earthquake length, (iii) the high seismic wave created from energy in the earth's crust which was calculated at 6.3 in the Richter scale. More factors are described in Figure 6. They identified based on the concept of *GravityFactor* introduced in our DM Metamodel. The *GlobalClimateChange* (Figure 6) is an instance of the *ComplexityFactor* (Figure 5) defined in the Earthquake model. The Earthquake Model identified that the combination between the *GravityFactor* and the *ComplexityFactor* are key factors that contribute to the impact of earthquake disasters generally.

The model shown in Figure 6 shows that as a result of the *ChristchurchEarthquakeDisaster* concept it affects the *ChristchurchAffectedElement* concept. The elements of the *ChristchurchAffectedElement* are the infrastructure (e.g: schools, shops, roads), the natural sites (e.g: rivers, parks, trees), the civilian society (e.g: Christchurch local communities) and people at risk at Christchurch (including many tourists). The Earthquake Model (shown in Figure 5) suggests few elements need to be considered if a comprehensive earthquake management is to be developed. These elements are the following: (a) DM policy (through *EarthquakePolicy*), (b) various DM resources (e.g: *EarthquakeResource*), (c) emergency rescue team (as *EarthquakeEmergency-Team*), (d) role of emergency actors (as *EarthquakeRescuerRole*), (e) DM procedure (as *EarthquakeProcedure*) and (f) DM requirement (*EarthquakeManagement-Requirement*). By instantiating all these earthquake concepts from the M1-level model, we utilized them into Christchurch

Figure 7 A Nuclear Disaster Model (M1-level), the second sample of model conformance from the DM Metamodel

Figure 7 presents the modeling structure of a Nuclear Disaster Model derived from its conformance of the DM Metamodel. The model describes the coordination of the nuclear disaster Objects derived from the concepts in our metamodel. These collection Objects represent the arrangement of DM elements that need to be handled of before, during and after the nuclear disaster strike. The model again verifies the capability of our metamodel into modeling a variety of other domain models. The model transformation process is executed in a similar manner to deriving the Earthquake Model in our first case. The model transformation between M2-level to M1-level allows a model to inherit as many relevant concepts as it necessitate for the purpose of its new disaster specific model.

In Figure 7, the Nuclear Disaster Model illustrates the arrangement of Objects corresponding to nuclear disaster scenarios. For example, in any nuclear disaster situation, the *HighRadiationExposed* becomes the concept which could distress the disaster affected elements (*NuclearDisasterAffectedElements*). To create a semantic link between both concepts, we tie by using the “*AffectedWellness*” relationship (refer Figure 7). The affected elements consist of people at risk, natural site, infrastructure and civilian society. The model specifies that *NuclearPowerPlantProblem* is the concept that creates the nuclear disaster problem. The disaster is also triggered by the *NuclearCoolingSystemFail*. For the purpose of organizing the model of this catastrophe, the nuclear DM requires a few sets of objects. These objects comprise of the *NuclearDisasterPolicy*, *NuclearDisasterProcedure*, *NuclearDisasterDomain-Knowledge* and a few other objects (refer to Figure 7 for the complete set).

In our approach, a DM Knowledge Repository can be developed from a database that stores the collection of previous disaster case solutions. Specifically for the DM Metamodel, each concept used in the artifact has its own *Procedure Fragment*. The fragment contains the detailed information about the concept including the concept operations, the concept attributes and the concept relationships. It itemizes all information of how the concept should work. For example, for a concept of *DisasterActionService*, the procedure fragment will have the information of what are the operations required by this concept, what are the relationships between the concepts and what are the requirements necessary for this concept.

```
(Metamodel) M2-Level: <<Events>>
(Model)      M1-Level: <<HighRadiationExposed>>
(User Model) M0-Level: <<FukushimaHighRadiationExposed>>
```

Decision fragment for Symptoms of Acute Radiation (within one day)

```
IF RADIATION = "0 – 0.25 Sv (0 – 250 mSv)" THEN SYMPTOM = "None"
IF RADIATION = "0.25 – 1 Sv (250 – 1000 mSv)" THEN SYMPTOM = "Some people feel
nausea and loss of appetite; bone marrow, lymph nodes, spleen damaged"
IF RADIATION = "1 – 3 Sv (1000 – 3000 mSv)" THEN SYMPTOM = "Mild to severe
nausea, loss of appetite, infection; more severe bone marrow, lymph node, spleen
damage; recovery probable, not assured"
IF RADIATION = "3 – 6 Sv (3000 – 6000 mSv)" THEN SYMPTOM = "Severe nausea, loss
of appetite; hemorrhaging, infection, diarrhea, peeling of skin, sterility; death if
untreated"
```

```

IF RADIATION = "6 – 10 Sv (6000 – 10000 mSv)" THEN SYMPTOM = "Above symptoms
plus central nervous system impairment; death expected"
ELSE
  RADIATION = "Above 10 Sv (10000 mSv)" THEN SYMPTOM = "Incapacitation and
death."

```

NOTES: Sv = Sievert (SAFE PUBLIC DOSE LIMITS for exposure from uranium mining or nuclear plants are at 1 mSv (miliSievert)/year)

Figure 8 The sample of decision-statement for the *HighRadiationExposed* concept. This illustrates how the decision making process is structured within the *Procedure Fragment*. In a real disaster problem, people who need to use the concept can automatically make a decision based on this decision-statement.

When a metamodel is transformed into its conformance model, the Procedure Fragments can also be derived by the model. Attached together with the Procedure Fragment is the *decision-statement* required for the concept to perform its activities. For example, Figure 8 displays a sample of the Procedure Fragment of the *HighRadiationExposed* concept for a specific task of deciding the symptoms of acute radiation to the affected. The decision is shown in a decision-statement form to make it look more user-friendly in view of the DM users. By providing decision support information through fragments, the model can support various decision making activities to each DM concepts in our metamodel. As a result, the decision approach supports many DM problems in its real world situation. We believe that by preparing the DM users with the information to support their decision, the time required for finding a disaster solution can be reduced significantly.

The *User Model* in Figure 9 represents the instance of one real nuclear disaster scenario in Fukushima, Japan. The *User Model* is an instance model of the Nuclear Model depicted by Figure 6. The result from the *FukushimaNuclearDisaster* concept is it affects the *FukushimaAffectedElement* concept. The elements in the *FukushimaAffectedElement* include people, natural site, civilian society and infrastructure of Fukushima. The Nuclear Model shown in Figure 7 suggests a few DM elements for managing the nuclear disaster problems. Some of these DM elements are the following: (a) Earthquake Policy (through *FukushimaEarthquakePolicy*), (b) Emergency team of nuclear disaster (as *FukushimaEmergencyTeam*), (c) various resource of nuclear disaster (as *FukushimaNuclearRescueResource*), (d) role of emergency actors (as *RescuerRole*), (e) DM procedure (as *EarthquakeProcedure*) and (f) DM requirement (*EarthquakeManagement Requirement*). By instantiating all these earthquake requirement concepts from *Model* at M1-level, we utilize them into the Fukushima Nuclear Model (M0-level).

4.0 Conclusion and discussion

The metamodel that has the capability to identify the DM concepts. It can also specify the DM modelling language. It describes DM concepts, the way they are arranged, related and constrained. To show the realization of these DM concepts as used in the metamodel, we demonstrate the conformance of various DM models from the metamodel. The DM Conformance Framework that underpins this is presented.

Through the framework, we illustrated how the DM Metamodel that is positioned in the M2-level in MOF Metamodelling Framework could be conformed to be the new DM Model in the M1-level. The results from the DM Model can then describe the *User Model* positioned at M0-level. The *User Model* is a target model of this approach that can describe the real scenarios of disaster management. In this paper, the 2011 Christchurch Earthquake disaster in New Zealand and the recent 2011 Nuclear Meltdown in the aftermath of the tsunami in Japan are chosen as two illustrative case studies. We believe we showed through a model-driven disaster management, the complexities of this domain can be structurally modeled. From the language that is described by the DM Metamodel, users of the DM can practice and understand their domain more effectively.

Generally, how many of concepts from the metamodel are reused depends on the context of the disaster and to what extent the DM expertise available needs to be complemented or supplemented. Even when it is partially reused, the details of the concepts or actions described by the metamodel, may still need to be refined by its users. The metamodel as such serves as a guide for knowledge sharing, enabling formulation of DM approaches as new situations arise.

References

- Aini, M. S., Fakhrol-Razi, A., Daud, M., Adam, N. M., & Kadir, R. A. (2005). Analysis of Royal Inquiry Report on the Collapse of a Building in Kuala Lumpur: Implications For Developing Countries. *Disaster Prevention and Management*, 14(1), 55-79.
- Alexander, D. E. (2002). *Principles of Emergency Planning and Management* USA: Oxford University Press, USA.
- Alhir, S. S. (2003). Understanding the Model Driven Architecture (MDA). 11(3), 17. Retrieved from <http://home.earthlink.net/~salhir>
- Asian Disaster Preparedness Center (ADPC). (2002). *Module 4: Capacity Building in Asia using Information Technology Applications (CASITA)*. Bangkok, Thailand.
- Aßmann, U., Zschaler, S., & Wagner, G. (2006). Ontologies, Meta-models, and the Model-Driven Paradigm *Ontologies for Software Engineering and Software Technology* (pp. 249-273): Springer Berlin Heidelberg.
- Beydoun, G., Low, G., Henderson-Sellers, B., Mouraditis, H., Sanz, J. J. G., Pavon, J., et al. (2009). FAML: A Generic Metamodel for MAS Development. *IEEE Transactions on Software Engineering*, 35(6), 841-863.
- Beydoun, G., Low, G., Mouraditis, H., & Henderson-Sellers, B. (2009). A Security-Aware Metamodel For Multi-Agent Systems. *Information and Software Technology*, 51(5), 832-845
- Davis, P. K., & Bigelow, J. H. (2002). *Motivated Metamodels*. Paper presented at the Proceedings of the 2002 Performance Metrics for Intelligent Systems Workshop (PerMIS'02), Gaithersburg MD, USA.

- Department of Education and Early Childhood Development (DEECD). (2010). Bushfire Information. Retrieved 18 March 2010, from <http://www.education.vic.gov.au/about/bushfires/default.htm>
- Greenfield, J., Short, K., Cook, S., Kent, S., & Crupi, J. (2004). *Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools*: Wiley Publishing Incorp.
- Heghe, H. V. (2011). *Knowledge Centric Management*: Ecademy Press.
- Henderson-Sellers, B. (2011). Bridging metamodels and ontologies in software engineering. *Journal of Systems and Software*, 84(2), 301-313.
- Hommes, B.-J. (2005). Evaluating Conceptual Coherence in Multi-Modeling Techniques. In J. Krogstie, T. Halpin & K. Siau (Eds.), *Information Modeling Methods and Methodologies* (pp. 43). USA: Idea Group Publishing (IGP).
- Jeusfeld, M. A., Jarke, M., & Mylopoulos, J. (2009). *Metamodelling for Method Engineering*. London, England: The MIT Press.
- Lalonde, C. (2011). Managing crises through organisational development: a conceptual framework. *Journal of Disasters*, 35(2), 443-464.
- Levendovszky, T., Rumpe, B., Schatz, B., & Sprinkle, J. (2010). Model Evolution and Management *MBEERTS* (Vol. 6100, pp. 241-270). Berlin Heidelberg: Springer-Verlag.
- M. Picka. (2004). Metamodelling and Development of Information System. *Agriculture Economics*, 2(50), 65-70.
- Nordstrom, G. G. (1999). *Metamodeling - Rapid Design and Evolution of Domain-Specific Modeling Environments*. Vanderbilt University, Nashville, Tennessee.
- OMG. (2002a). Meta Object Facility (MOF) Specification (1.4 ed.): Object Management Group.
- OMG. (2002b). MOF 2.0 Query / View / Transformations RFP. from <http://www.omg.org/docs/ad/02-04-10.pdf>.
- Othman, S. H., & Beydoun, G. (2010a). *Metamodel-based decision support system for disaster management*. Paper presented at the The 5th International Conference on Software and Data Technologies (ICSOT'2010).
- Othman, S. H., & Beydoun, G. (2010b). Metamodelling Approach Towards a Disaster Management Decision Support System. In Leszek Rutkowski et. al (Ed.), *Artificial Intelligence and Soft Computing* (pp. 614-621). Zakopane, Poland: Springer-Verlag, Berlin.
- Seidewitz, E. (2003). What Models Mean. *IEEE Software*, 20(5), 26 - 32
- Stahl, T., Voelter, M., & Czarnecki, K. (Eds.). (2005). *Model-Driven Software Engineering, Technology, Engineering, Management*: John-Wiley & Sons Ltd.
- Trabelsi, C., Atitallah, R. B., Meftali, S., Dekeyser, J.-L., & Jemai, A. (2011). A Model-Driven Approach for Hybrid Power Estimation in Embedded Systems Design. *EURASIP Journal on Embedded Systems*, 2011(Article ID 569031), 15
- Varro, D. (2004). *Automated Model Transformations for the Analysis of IT Systems*.
- W3C Incubator Group. (2008). Emergency Information Interoperability Frameworks. Retrieved Nov 2009, from <http://www.w3.org/2005/Incubator/eiif/XGR-Framework-20090806/#ack>