

University of Wollongong

Research Online

Faculty of Engineering and Information
Sciences - Papers: Part A

Faculty of Engineering and Information
Sciences

1-1-2013

DM model transformations framework

S Othman
sho492@uow.edu.au

G Beydoun
University of Wollongong, beydoun@uow.edu.au

Rodney Clarke
University of Wollongong, rclarke@uow.edu.au

S Opper

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



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

Recommended Citation

Othman, S; Beydoun, G; Clarke, Rodney; and Opper, S, "DM model transformations framework" (2013).
Faculty of Engineering and Information Sciences - Papers: Part A. 1324.
<https://ro.uow.edu.au/eispapers/1324>

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

DM model transformations framework

Abstract

Metamodelling produces a 'metamodel' capable of generalizing the domain. A metamodel gathers all domain concepts and their relationships. It enables partitioning a domain problem into sub-problems. Decision makers can then develop a variety of domain solutions models based on mixing and matching solutions for sub-problems identified using the metamodel. A repository of domain knowledge structured using the metamodel would allow the transformation of models generated from a higher level to a lower level according to scope of the problem on hand. In this paper, we reveal how a process of mixing and matching disaster management actions can be accomplished using our Disaster Management Metamodel (DMM). The paper describes DM model transformations underpinned by DMM. They are illustrated benefiting DM users creating appropriate DM solution models from extant partial solutions.

Keywords

model, dm, transformations, framework

Disciplines

Engineering | Science and Technology Studies

Publication Details

Othman, S., Beydoun, G., Clark, R. & Oppen, S. (2013). DM model transformations framework. 24th Australasian Conference on Information Systems (ACIS 2013) (pp. 1-10). Melbourne: RMIT University.

DM Model Transformations Framework

Abstract

Metamodelling produces a ‘metamodel’ capable of generalizing the domain. A metamodel gathers all domain concepts and their relationships. It enables partitioning a domain problem into sub-problems. Decision makers can then develop a variety of domain solutions models based on mixing and matching solutions for sub-problems identified using the metamodel. A repository of domain knowledge structured using the metamodel would allow the transformation of models generated from a higher level to a lower level according to scope of the problem on hand. In this paper, we reveal how a process of mixing and matching disaster management actions can be accomplished using our Disaster Management Metamodel (DMM). The paper describes DM model transformations underpinned by DMM. They are illustrated benefiting DM users creating appropriate DM solution models from extant partial solutions.

Keywords

Metamodelling, Modelling Language, Metamodel, Disaster management, Model Transformation, Instantiation

INTRODUCTION

Disaster Management (DM) involves various elements such as, authorities, organizations, processes, people and coordination in the domain. It is a complex process that engages with many unpredictable situations in its domain. Many DM problems have a few collection of successful solution. But, because of all these successful solutions have not been properly recorded, many of this information can be lost or scattered. If all the information is stored appropriately, the time required for domain users to solve the recurrent problem will be significantly reduced. A metamodel serves as a powerful tool to share knowledge with people outside the studied domain. Our metamodel that describes a common language for DM is called the Disaster Management Metamodel (DMM). Work that we present in this paper is the extended result from the DMM that we have produced in (Othman and Beydoun 2010; Othman and Beydoun 2011). In relation to result of DMM in that paper, we have provided the detailed accounts of actual development of the metamodel. DMM consists of four (4) classes of concept which include a *Mitigation, Preparedness, Response* and *Recovery* set. All problems and solutions in DM were revealed by identifying all existing key concepts appeared in 30 investigated DM models. All identified concepts were partitioned and structured into the DMM. The DMM uses the 4-layer of Meta Object Facility (MOF) framework offered by the Object Management Group (OMG 2002). The use of this metamodelling framework offers many benefits to its user. One of the benefits of DMM is, it can be manipulated as a *domain modeling template* by many DM users. From here, they can instantiate a variety kind of DM solution model based on the disaster problems. Besides, the template also provides a solution for a range type of disaster. In this article, we reveal how a transformation of model from the DMM can construct a new DM solution model through a process called *Instantiation*. From this process, users of the metamodel can gather input of what are the data they need for their new solution model (Lagerstrom, Johnson et al. 2010).

The paper also extends the model transformations illustrated in (Othman and Beydoun 2013). To demonstrate this process, this paper introduces a *system prototype*, a Metamodel-based Disaster Management Decision Support System (DMDSS). With the potential in finding quick decision solutions based on a semantic model and domain knowledge constructed by the DMM, we believe that DMDSS can assist in reducing the amount of time required for decision and mobilisation process in disaster situations (Oppel, S et al. 2009). DMM by itself provides a bird’s eyeview of DM processes and how they can interlink. This alone can assist many users (e.g.: Emergency Service teams including Incident Controllers, Planning, Situation and Intelligence Officer users, Local and State Government, Other Emergency Managers, Aid Agencies, Communities, people at risk (e.g. in a neighbouring region of a flood zone)) in the domain to perform their DM tasks in a more structured view before, during or after a disaster had happened.

The rest of this paper is structured as follows: The first part of the paper overviews metamodelling, the main technique used for this research. The second part describes a Model Transformation Process in the metamodelling environment which includes vertical and horizontal dimensions of transformation. The second part also presents the Preparedness-phase class of concepts as the sample of Disaster Management Metamodel to be used. In the third part, we present an “*Instantiation*” as one of a vertical model transformation in the DM metamodelling environment. Finally, we conclude the paper with a discussion on the findings and future works of this research.

BACKGROUND ON METAMODELLING

A metamodel is the conceptual foundation of a ‘modeling language’ which in turn enables domain users to abstract, share knowledge and makes it easier to solve problems using a model as a starting point (Seidewitz 2003). Various kinds of modelling languages have been developed for a variety of knowledge disciplines. A model is an instance of a metamodel which represents a *phenomenon from the real world*. Models for a specific domain, or *Domain Models*, contain abstraction of domain concepts. They often describe human interactions. To create a domain model understanding of domain concepts and their relationships is required. This understanding can be encapsulated in the semantics of a language of the domain and can be developed accurately using metamodeling. In DM, we develop a language to model concepts in DM (e.g.: *Emergency Services Team*, *Early Warning*, *Supporting Agencies* or *Team Coordination*). We study and identify the relationships among DM activities and teams (e.g.: (i) what ‘*Evacuation Plan*’ has to do with ‘*People at Risk*’? or (ii) what are the relationships that exist between *Supporting Agencies* and *Emergency Services Teams*?). We applied a metamodeling framework, MOF (Modelling Objects Framework) (OMG 2002) which has 4 (four)-level of modeling hierarchical architecture in its language modeling framework (Table 1): The **M3-Level** reserved for *Meta-metamodel* elements describing the structure and semantics of the language specifying metamodels; The **M2-Level** reserved for a *Metamodel* layer (an instance of a meta-metamodel) describing the structure and semantics of language for specifying a model (e.g., *MetaClasses*, *MetaAttributes*, *MetaOperations*, *MetaRelationships*); The **M1-level** describing *Models* (instances of a metamodel) and comprising of the metadata that describe data in the information layer; the lowest level in MOF framework is the **M0-Level** or the *User Model* is an instance of a Model. M0-level can also be called as an information layer in MOF. In our DM level, an M0-level covers information of a specific disaster and represents the real-world DM activities performed by DM users.

Table 1. MOF Four-Layer Elements Description (extended from Frankel 2003)

Model Level	Level Name	Description	Elements
M3	Meta-metamodel	MOF Layer <ul style="list-style-type: none"> Infrastructure for a metamodeling architecture Defines the language for specifying metamodels 	<ul style="list-style-type: none"> MetaClass, MetaAttribute, MetaOperation MOF::Class, MOF::Attribute, MOF::Operation
M2	Metamodel	An instance of a Meta-Metamodel <ul style="list-style-type: none"> Defines the language for specifying a model Consists of MetaClasses, MetaAttributes, MetaOperations, MetaRelationships 	<ul style="list-style-type: none"> Metamodel Fragment (M2 Fragment) E.g.: <i>Evacuation</i>, <i>EmergencyPlan</i>, <i>EarlyWarning</i>
M1	Model	An instance of a Metamodel <ul style="list-style-type: none"> Models a specific information domain 	<ul style="list-style-type: none"> Data Model Fragment (M1 Fragment) E.g.: (<<Evacuation>> <i>EvacuationRoute</i>, <i>EvacuationPlan</i>, <i>PeopleToEvacuate</i>)
M0	User Model	An instance of a model <ul style="list-style-type: none"> Models a user real world data 	<ul style="list-style-type: none"> User Model Fragment (Real World Data) E.g.: (samples from previous Flood Evacuation Cases such as in the 2001 Grafton flood in Australia), (Pfister, N 2002).

As an example, a problem in DM is coordinating evacuation processes during a disaster. In our metamodel (Figure 2), we use the *Evacuation* concept to represent the evacuation activity in the DM domain. This concept appears in most DM models. In DMM, the *Evacuation* is defined as ‘*The planned relocation of persons from dangerous or potentially dangerous areas to safer areas and eventual return*’ (EMA 1998). The abstraction of concepts situated in higher level is relatively fixed because of the generalization value of the concept (Gharehdaghi 2003; Beydoun and Hoffmann 1998). This means that the definition of a general concept used in a metamodel is likely to be more stable than instances of the concept used in lower level. For problems that share the same concept operation such as a flood evacuation, an earthquake evacuation or a nuclear meltdown evacuation, DMM will only use a general term of ‘*Evacuation*’ to represent these three ‘similar task-concepts’ in our metamodel (at M2-Level). A database schema based on the metamodel will remain much more stable than any conceptual schema developed using concepts from lower levels of abstraction. DMM can then be used to generate multiple views of the same domain concept for servicing different DM users. For example, the fragment for *Evacuation* will contain all attributes and operations that illustrate the semantic of the concept. Example attributes are: *evacuationRoute*, *evacuationWarning-Time*, *evacuationVehicle-Movement-Time*, *evacuationCentre*. Example operations are: *evacuationDecision()*, *evacuationMobilise()*, *evacuationWarning-Delivery()*, *evacuatePeopleAtRisk()*, *setupAndorganize EvacuationCentres()*. By taking the example of *evacuatePeopleAtRisk*, one of the concept operations enclosed in the *Evacuation*, it can be seen that the same concept operation is required to be performed by more than one group of user in real-world disaster scenario. Consider the scenario of a flood that travels down a catchment. Prior to the flood occurring in the *Preparedness Phase* an *Emergency Services Team* will *assess and plan* a range of the evacuation attributes such as the *Decision*, *Mobilise* and *Warning* tasks for *People at risk* in a number of management sectors across a community.

This process informs teams during the *Response Phase* when a flood event is occurring who will implement the required actions(operations) in each sector and the community as a whole. The same tasks also need to be executed for the multiple sectors in each additional community containing *People at Risk* further downstream in the catchment. In fact the same task set is required to be replicated across multiple catchments or even the State that may experience flooding at the same time. Based on these scenarios, we can see how there are different tasks (*assessandplanEvacuation* and *evacuatePeople*) replicated multiple times which come from the same DM concept (*Evacuation*). But the way these scenarios need to be implemented is different based on who the stakeholders are. In DMM, we structure this complexity of DM processes through the use of concept fragments in a proper way of representation.

DISASTER MANAGEMENT METAMODELLING

The relationships between metamodels and models are described through *model transformation* (Vytautas Stuiikys 2010). Model transformation is a process of converting one model to another model (OMG 2003). The presence of a metamodel gives a capability to users in the domain to create various solution models based on the elements as structured in the metamodel. Model transformation in MOF metamodeling can be viewed in *vertical* and *horizontal* dimensions (Bieman 2001). In *horizontal transformation*, the dimension involves transforming a model into a target model which is in the *same level* of abstraction as a model, no matter high or low is the artefact level. The transformation of this dimension presents the *evolution* of artefacts. A *vertical transformation* presents the transformation of model at *different level* of abstraction level. *Instantiation* of model from higher level to a lower level of abstraction is the example of this kind of transformation. Instantiation is performed when a model instance is created from its metamodel (Vytautas Stuiikys 2010). The vertical transformation of DMM to lower DM models is the chief aim of the transformations we demonstrate in this paper. This not only underpins the feasibility of a DMM knowledge repository, but also the access to this knowledge by future users. The key issue in the instantiation process is how to represent a *Model Instance* from level M2 to level M1. There are two choices: One is to represent an instance model *exactly* as its conformant metamodel (Brottier, Fleurey et al. 2006). This is referred to as *Full Instantiation*. The second choice is a more *human-friendly* model representation of a metamodel. Some relationships shown in a metamodel are hidden behind the model without highlighting the conformance to its metamodel. This process is called *Partial Instantiation*. In our use of DMM detailed in the next section, we provide both instantiation techniques.

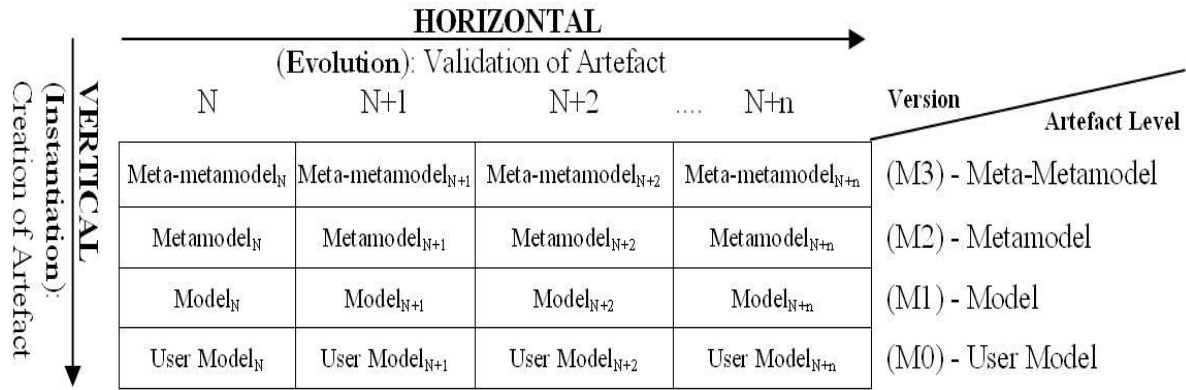


Figure 1: Horizontal and Vertical Model Transformation in MOF Metamodeling

To develop DMM, we observed a total of 30 investigated models of DM where the details of this analysis we detailed in (Othman and Beydoun 2010). These observations showed us that the DM domain has four phases or classes of concepts: *Mitigation*, *Preparedness*, *Response* and *Recovery*. Each phase has different processes and stakeholders. DMM generalizes and structures DM concepts according to the phases. For example, concepts required for the DM during the *preparedness* phase of a disaster are presented by the Preparedness-phase class of concepts of DMM shown in Figure 2. The arrangement of all DM concepts is important and crucial because from here DM users will get a clearer picture of what are activities involved in each DM phase. In this paper, we use Preparedness as a sample of our DM phase where many elements in the set were utilized in next section.

We defined Preparedness as a phase where ‘the knowledge and capacities developed by governments, professional response and recovery organizations, communities and individuals to effectively anticipate, respond to and recover from the impacts of likely, imminent or current hazard events or conditions’ (UNISDR 2009). Some key concepts that should be generated during this phase include *Evacuation*, *Early Warning*, *Monitoring*, a *Public Media* or a *Preparedness Plan*.

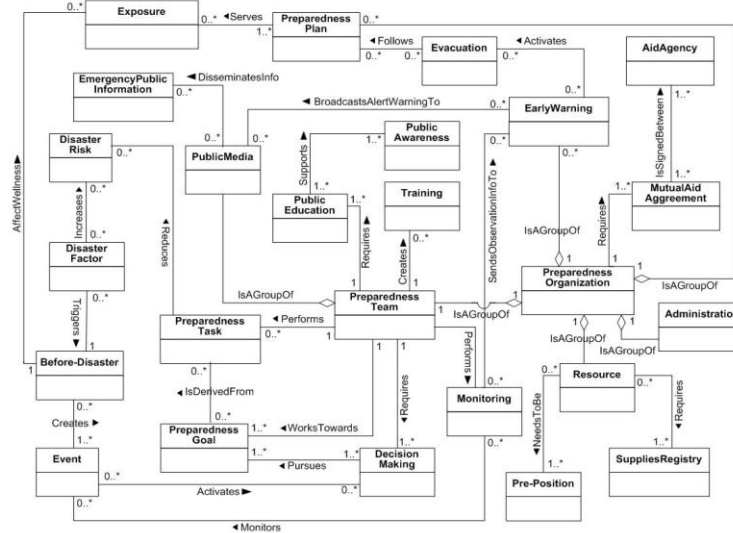


Figure 2: A *Preparedness*-phase class is one-of-four set of class of concept in the Disaster Management Metamodel (DMM). This contains DM concepts and relationships that must appear during the preparedness phase of DM in any disaster scenario.

DM DSS USING MODELS TRANSFORMATION

We guide the DM users with the transformation of models through our DM knowledge management system prototype. We call this system DM Decision Support System (DMDSS). This section presents the models transformations in DMDSS underpinned by a metamodeling environment.

Our prototype system design allows users to instantiate various DM models from the DMM metamodel. In DMM Instantiation framework (Figure 3), target users include the local/state/country authorities, the monitoring users or the emergency coordinators. The system also caters for newcomers to the DM domain such as researchers who want to find a solution for certain disaster problem. They will be asked to provide information for the disaster problems that they need to overcome (*Disaster Problem Input*). DMDSS will receive new input from users. Through the '*Model Formulation Decision Tool*', the DMM library of rules and DMM library of notation will be triggered. An example rule, is the following instantiation rule:

Rule Syntax : $C_M(\mu_{DMM})$

Rule Meaning : Mitigation Concepts of the Disaster Management Metamodel

Rule Construct : MitigationConcept::= (<MitigationDMMClass AND MitigationM2Fragment>)

The rules corresponding to every DMM concept will identify what is the best DM concept combination for constructing the solution model. The combination of concepts is derived based on the inputs given by the users. For an example, if a *Disaster Monitoring User*, Y wants to find a new solution model that can help them to solve a problem in flood monitoring, the system will require further information e.g. (i) Type of disaster: (input1=*Flood*), (ii) Type of Domain Stakeholder: (input2=*Monitoring User*) and a other requirements. These inputs are important before the system can proceed to the next step. Rules from the DMM library will then used to analyse inputs from this user. A library of Rules is used for informing the system of what are the components they need to process before a new solution model component is used.

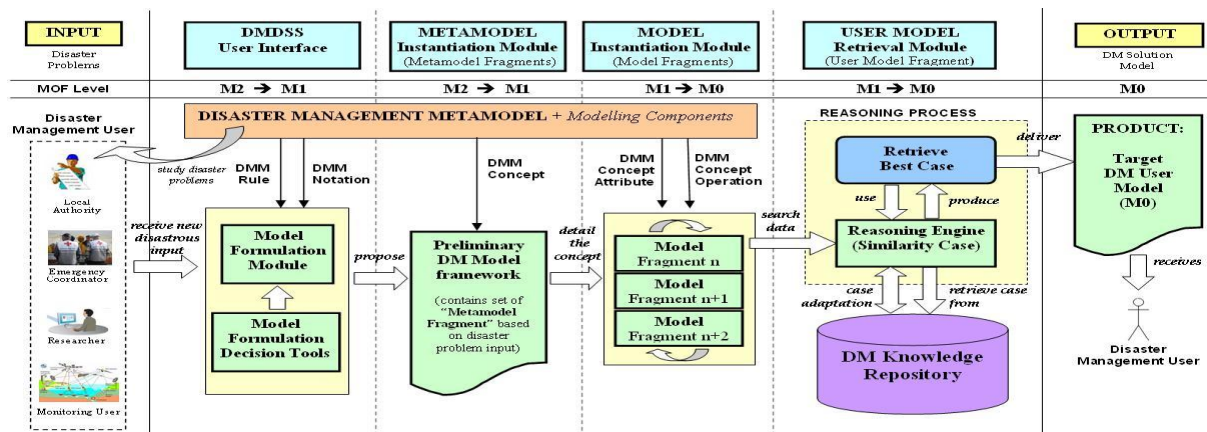


Figure 3: DMM Instantiation Framework

Instantiation M2-level (DMM) to M1-level (a DM Model)

The system needs to identify combinations of DM concepts to develop the solution model. The requirements associated with each of the concepts need to be first identified. We use the term ‘*Metamodel Fragment*’ to represent the components of Model unit for each metamodel concept. The metamodel fragment at M2-level contains concept relationships, attributes and operations. *Concept Relationship* contains information of other DM concepts that are related to the metamodel concept. *Concept Attributes* contains necessary information the metamodel concept requires in order to perform tasks of the concept (e.g.: for *Evacuation* concept, attributes could be *evacuationRoute* or *evacuationLocation*). *Concept Operation* contains information of the DM activities, actions, tasks of the metamodel concept (e.g.: for *Evacuation* concept, operation could be *evacuatePeopleAtRisk*, *performEvacuationPlan* or *setupEvacuationCentre*). Say for a user Y, system will identify ‘*Monitoring*’ as one of important the concept that the system need to formulate (refer Figure 1 for the position of *Monitoring* in our metamodel). Output from a Model Formulation Module will be a collection of relevant concepts for developing the solution model. To perform a monitoring process of disaster, all concepts that have any relationship with the *Monitoring* need to be identified (e.g.: *EarlyWarning* concept has a ‘*SendObservationInfoTo*’ relationship with *Monitoring*). At this stage, a model is still in its M2-level of metamodeling layer. Only after the Model Formulation Module has proposed a *Preliminary DM Model* then the model is transformed to the M1-Model in our system. In Figure 4, we outline a sample of instantiation in a multi-level view of the modelling layer specifically for a *Monitoring* concept. As an example, model M1 can extract all important operations for a monitoring process of a disaster as defined by the *Monitoring* fragment in M2-level. The *monitorEvent* and the *analyseEvent* are two concept operations for modelling the disaster monitoring activity in M1-level. *monitorEvent* for example, will be a fragment of a Model used to instantiate to another User Model in M0-level. *monitorFloodEvent* or *monitorBushfireEvent* would then appear in the M0 model to represent real world implementation of the concept fragment.

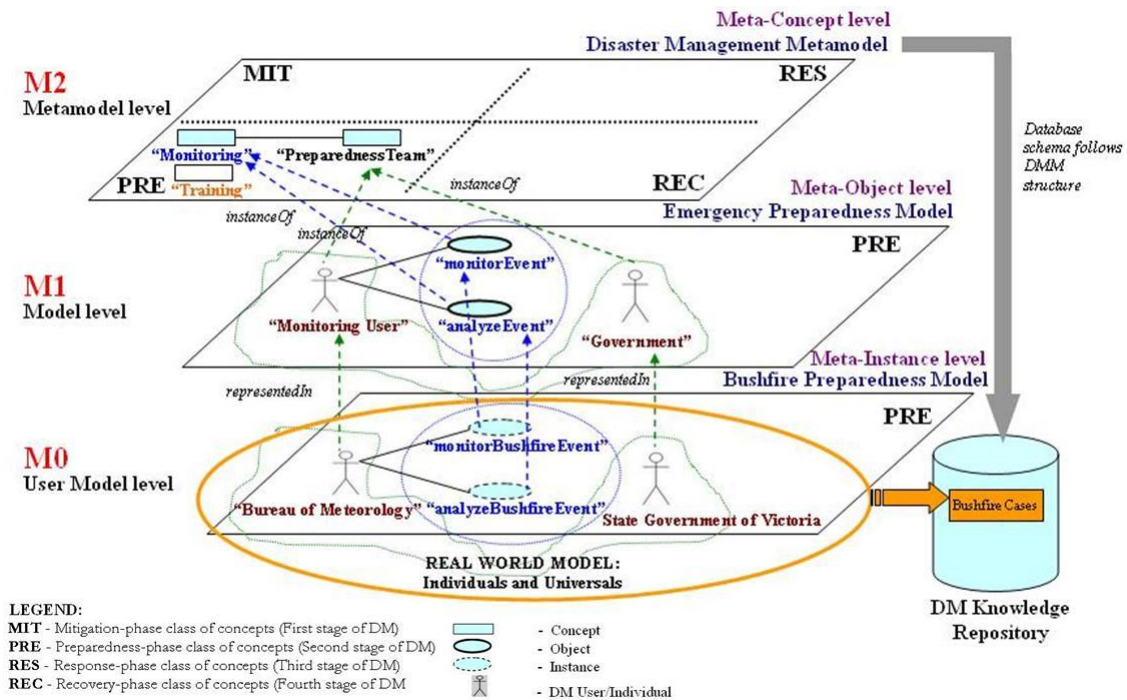


Figure 4: Instantiation of DM concept (*Monitoring*) at multiple level of abstraction

A ‘*Preliminary DM Model*’ is a framework of M1-Model which contains all DM concepts for the solution model. A model that is constructed at this level will present this framework. The framework is formed based on inputs of the disaster problems entered by the user. We represent a sample of the *Preliminary DM Model* (I2 in Figure 5 to illustrate the interface of this model) through an *Emergency Preparedness Model*. In this sample, we use an *Evacuation* as a sample of a metamodel fragment (refer I1 of Figure 5) where the operation of this concept are modelled through I2. All important attributes and operation in the metamodel fragment of the *Evacuation* concept will be instantiated by M1-model in I2.

Instantiation from M1-level (a DM Model) to M0-level (User Model - Real World Model)

The manipulation of concept attributes, concept operations and concept relationships in *Model Instantiation* module will create M1-Model. M1-Model at this stage will have few fragments where each of the fragments

describes information of User Model in level M0. The fragment at M1-level is called a *Model Fragment*. The circle dotted line **A** as shown in Figure 5 shows the Model Fragment of *Monitoring*. In A, (i) <<Monitoring>> refers to a Metamodel concept of *Monitoring*. The concept operation is sampled through ‘analyse events’. The attributes and relationship for this model fragment is hidden behind the model fragment button. After all model fragments are identified, the user will be brought to the next module of a *User Model Retrieval* module. At this stage, DMDSS will retrieve the most suitable model fragments formed in the Model Instantiation Module. All the retrieved information will be used as data for all model fragments in User Model level. Data in a User Model of M0-level will represent a real solution model to be implemented in the real disaster application.

Figure 6 shows a sample of activity for the ‘*PreparednessTask*’, a concept used in the Preparedness-phase of DMM. M1 Fragment in Figure 6 shows the ‘*establish coordination arrangement*’ activity (shown by **B2**) which is a concept operation in ‘*PreparednessTask*’. The collection of best cases retrieved will appear in M1 fragment fields. The result of the cases could provide the solution sought by the user. The case shown in Figure 6 is a sample of a similar case that is retrieved based on previous knowledge stored in the repository. For example, the source of the retrieved case in the above figure is from Australia Emergency Management (shown by **B3**). Users who can implement this model are agencies at federal, state, district or local level as shown in ‘(M0) *User Model Level*’ field (shown by **B4**). An M1 fragment will also provide the details of the activity tasks that can be performed by all the various users through ‘*DM Activities*’ field (shown by **B5**). Other than presenting activity tasks, an M1 fragment also provides data to support decision making for the specific activity task. Through the *Decision Fragment* (shown by B6) field, we guide the user in making a decision for the activity task as presented by the M1 fragment. Representing decision in this way will be much more readable and human-friendly to the DM users. And, the output from our DMDSS will be the representation of Target DM User Model in M0-level of abstraction. The model will represent the real-world DM solution model that can be put into practice by the DM Users.

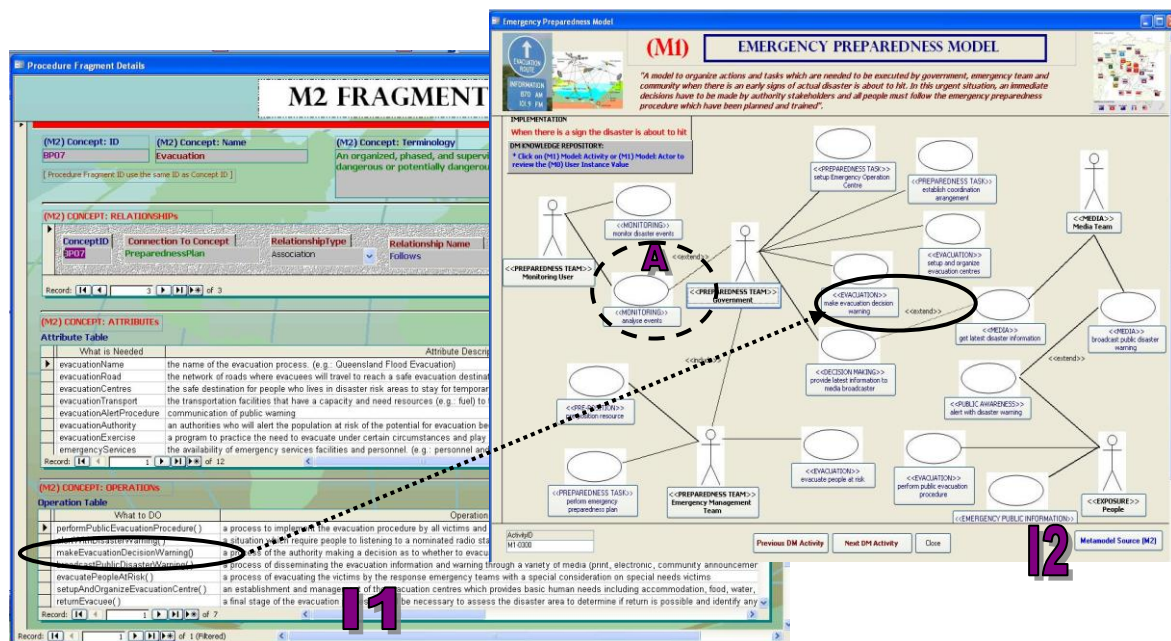


Figure 5: A snapshot of system interfaces of DMDSS. (a) The M2 Fragment of the *Evacuation* concept
(b) The Preliminary DM Model Framework of the *Emergency Preparedness Model*

Toward Deploying DMDSS in Real World Application

A common DM misconception is the belief that DM is separate from normal governance and community business. DM actually does not only involve emergency management agencies and response and recovery efforts for a disaster. DM should extend across all government sectors, non-governmental organizations/industry, from international levels to state and local levels. Extending DM knowledge needs to include individuals and every facet of society. People at risk for example, are a critically important group of DM users who normally face the problems during the disaster. We must be aware that in DM, everybody has some role to play in a DM cycle. This work tries to elucidate the responsibilities (tasks, roles, functions) to be carried out by various groups in DM. Figure 8 presents some of the possible models which can be derived as a solution model from DMM. Even, if one model (M1) can be instantiated from the metamodel, various other real-world solution model (M0) can be

derived from that M1 model. This is possible because of the integration of DMM with the domain knowledge repository.

Figure 6: A snapshot of system interfaces of DMDSS. (a) The M2 Fragment of the *Evacuation* concept (b) The Preliminary DM Model Framework of the *Emergency Preparedness Model*

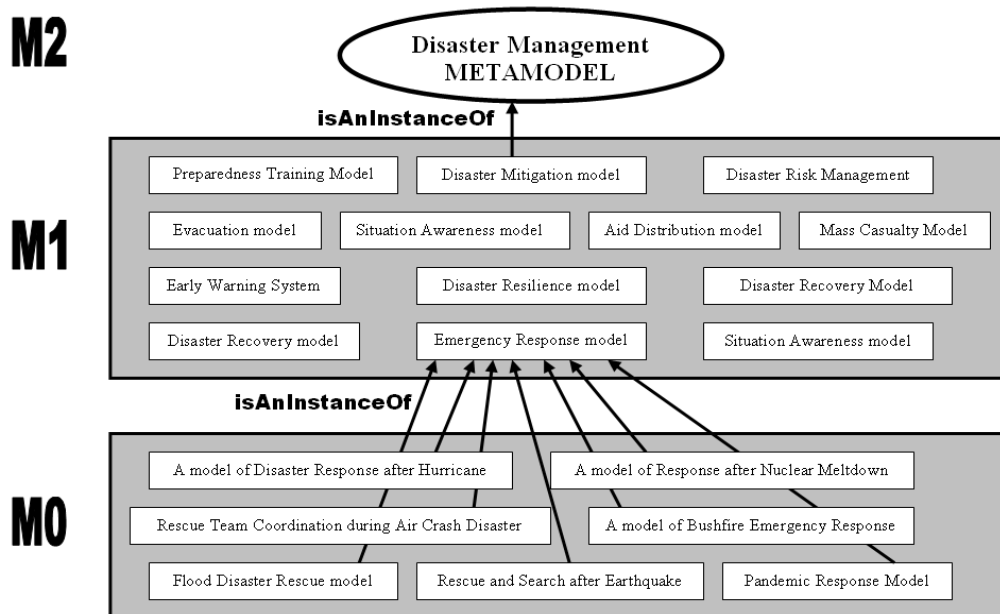


Figure 7: A sample of possible Models (M1) and User Models (M0) instantiated from DM Metamodel (M2)

Below are some of the disaster problems cases which are faced by different kind of users in DM domain. With the capability offered by DMDSS, the solution to these problems can be solved more effectively.

- (a) *Case Study 1*: W is a *Emergency Services Minister* of a State in Australia. She wants the DMDSS to help her make decisions on which steps she needs to take to support the *Recovery Phase* for a flood based on the latest situation of *flood* consequences in her state. The solution model must provide her with a similar case of how actions were taken during the past flood events or in other States.

- (b) *Case Study 2: X* is the *State Operations Controller* for the combat agency responsible for flood emergency management in a State in Australia. He wants to know what the best model solutions for a Major Flood Evacuation Plan. The model must present the coordination of different teams at State, Regional and Local levels to perform their functions during a flood event. In addition, the solution model also provides the best flood intelligence data, a flood emergency plan and incident action plan for each level of the incident management teams to follow.
- (c) *Case Study 3: Y* is a *Community Member* who lives in a flood-prone area in Wollongong, Australia. He wants to know what the best procedures to follow are if he is observing very heavy rainfall and rapid river rises in his area. He also wants the system to give him information and solution of what are the action he needs to take. The model must provide him with a specific information on where to seek information and warnings from authorities and when and where to evacuate to.
- (d) *Case Study 4: Z* is a *Country Representative* from France Government. His country wants to deliver some donation in form of humanitarian aid to Japan, a country that suffered from a tremendous Nuclear Meltdown and Tsunami disaster recently. The model that he requires must provide the process flow on how aid distribution can be delivered. He also wants to know what are the types of humanitarian aid suitable with this kind of tragedy.

To populate the repository from text based description to support the above scenarios, an unorthodox approach to understanding and capturing DM concepts will be employed. Complex organisations- including those involved in DM as described above- comprise functionally distinct professional groups that utilise different *profession oriented languages* (Nygaard 1988, 21) to express their understanding about the world and its disasters. Each of these professional groups can be distinguished by means of the situational languages they employ. This will refine concepts from DMM at M2. These situational and profession-specific languages are referred to as *registers* in the sociolinguistic literature (Eggins 2004, 85-112). Each DM group (for example planning, operations, coordination and support) are likely to have distinct registers. A notation called the *system network* can be used to represent the superordinal relations between situationally-defined *indexical lexical items* (Eggins 2004, 196-198; 107); those words that are agreed by a given group to describe the situations that are of interest and relevance to them. It is by taxonomic organisation of indexical lexical items that we can define a given register or distinguish between registers. The semantics of systems networks can be partitioned into *metamodel fragments* and *concept relationships*. Another group of sociolinguistic resources that are of particular use are referred to as *canonical genres*. These resources refer to the particular rhetorical staging required in order for a given communication to be relevant organisationally. Canonical genres are represented as directed cyclical graphs and they have been used to directly represent business processes and services in a number of related agent-oriented applications (see Clarke et al 2007; Clarke and Krishna 2006). Canonical Genre resources are used in two related ways within the DM Model Transformations Framework. They can be used in an analytical way by creating an *elicitation methodology* to recover domain specific knowledge for disaster stakeholders. The main idea here is that domain specific knowledge can be specified using a relatively small set of standardised rhetorical structures. Strategies exist for probing, repairing and realigning participants as they are encoding their knowledge into these culturally available and standardised structures. Researchers can also use their knowledge of these structures to test for sequencing and completeness of the represented knowledge (see Clarke 2006; Krishna et al 2006). Canonical Genres can be used to design appropriate *user interfaces* between the Input → DMDSS stage and the USER MODEL Retrieval Module → DM Solution Model stage (see Clarke 2001; Al Mansour 2012). This will involve replacing DM activities descriptions from various sources (e.g. web page content, DM manuals, etc ..) with canonical genres.

CONCLUSION AND FUTURE WORKS

This paper presents the transformation of models in metamodeling technique. It involves the *Instantiation* of Metamodel-level M2 to models (a Model-level M1 and a User Model-level M0) of its lower level of abstraction. From the instantiation process, potential of metamodel in solving the complex activities in the domain is demonstrated. A Disaster Management (DM) is chosen as our studied domain where language of this realm is modelled through a metamodel called a Disaster Management Metamodel (DMM). Unlike existing models, DMM is disaster independent. It offers an abstract view of the domain which can be used to unify extant models. The objectives of the development of DMM are to formulate a special tool, which has a potential to be an effective platform for sharing and integrating DM knowledge from varying sources which allows an interoperability of DM solutions and effective transfer of knowledge across the levels of emergency management agencies and other supporting agencies, between jurisdictions or even international boundaries. Through a prototype of Metamodel-based Disaster Management Decision Support System (DMDSS), we presented how an integration of metamodel (DMM) with its repository of knowledge domain (the DM Knowledge Repository) can provide better support for decision making to many users in the DM domain. To complement the metamodel, a

collection of rules and notations assist users on how to utilize the metamodel and find a solution based on the disaster problems encountered.

We believe by a formulation of domain knowledge through a metamodeling technique, the operations and information of how people fulfil their role and participate in their domain can be executed in a better way. Besides helping other people in the same domain to find a solution for a problem (recurrent situation), a development of this tool could also allow the process of checking the completeness of domain. As an example, in the advance of satellite technology, a new process X is introduced for the purpose of monitoring a disaster. As X is a new method in the monitoring process of disaster, the way monitoring users should perform their task for monitoring disasters also needs to be adjusted. Therefore, the concept of '*Monitoring*' that is used in the earlier version of our metamodel, needs to be validated to suit the improvement to the disaster monitoring process in the real world application. Thus, if any new concept emerges in the domain where the metamodel is studied, a validation process runs against the metamodel to identify the missing concepts in the metamodel. This assists the metamodel in checking and maintaining the completeness of the domain it models. Finally, once deployed continuous maintenance and evaluation of the concepts in the metamodel is required. Recent dynamic evaluation algorithms, as presented in (Beydoun et al 2013a; Beydoun et al 2011; Beydoun et al 2013b), and specifically tailored to evaluate individual concepts in a changing models will be employed.

REFERENCES

- Al Mansour, M. A. A. (2012) A Systemic Semiotic Analysis of Cultural Differences between Australian and Saudi eBusiness Websites *Unpublished PhD Dissertation, School of Management and Marketing, University of Wollongong*
- Beydoun, G., Hoffmann, A. (1998), Simultaneous modelling and knowledge acquisition using NRDR. 5th Pacific Rim Conference on Artificial Intelligence (PRICAI98). Springer. Singapore.
- Beydoun, G., García-Sánchez, F., Vincent-Torres, C.M, Lopez-Lorca, A.A., and Martínez-Béjar, R. (2013), Providing metrics and automatic enhancement for hierarchical taxonomies, Information Processing and Management, 49(1): 67-82.
- Beydoun, G., Lopez-Lorca, A., Garcia-Sanchez, F., Martinez-Béjar, R. (2011), How do we measure and improve the quality of a hierarchical ontology, Journal of Systems and Software 84 (12) (2011) 2363-2373.
- Beydoun, G., Hoffmann, A. (2013), Dynamic evaluation of the development process of knowledge-based information systems, Knowledge and Information Systems, Springer, 2013, 35 (1), 233-247.
- Bieman, R. F. a. J. M. (2001). Multi-View Software Evolution: A UML-based Framework for Evolving Object-Oriented Software. Proceedings International Conference on Software Maintenance (ICSM 2001), France.
- Brottier, E., F. Fleurey, et al. (2006). Metamodel-based Test Generation for Model Transformations: an Algorithm and a Tool. 17th International Symposium on Software Reliability Engineering (ISSRE 2006), Raleigh, North California, USA.
- Clarke, R. J. (2001) From System to Text: Documenting Computer Applications using Genre 3rd International Conference on Enterprise Information Systems (ICEIS) Proceedings Volume 2 Setúbal, Portugal, July 7-10, pp. 774-778
- Clarke, R. J. (2006) The Work that Analysts Do: A Systemic Functional Approach to Elicitation in Nilsson, A. G.; Gustas, R.; Wojtkowski, W.; Wojtkowski, W. G.; Wrycza, S. and J. Zupančič (2006) Advances in Information Systems Development: Proceedings of the 14th International Conference on Information Systems- Development Bridging the Gap between Academia and Industry Volume 1 NY: Springer Science + Business Media ISBN-10: 0-387-30834-2, pp. 325-336
- Clarke, R. J.; Ghose, A. K. and A. Krishna (2007) Systemic Semiotics as a basis for an Agent-Oriented conceptual Modelling Methodology in Gudwin, R. and J. Queiroz eds/ (2007) Semiotics and Intelligent Systems Development Hershey, PA: Idea Group Inc., Chapter 9, pp. 256-285 ISSN: 1-59904-063-8
- Clarke, R. J. and A. Krishna (2006) Embedding Social Processes into Agent-Oriented Conceptual Models Action in Language, Organisations and Information Systems (ALOIS 2006)- Fourth International Conference, November 1-2, 2006, Borås, Sweden, pp.59-73
- Eggins, S. (2004) An Introduction to Systemic Functional Linguistics 2nd Edition New York: Continuum
- Emergency Management Australia (EMA). (1998). Manual 3 Australian Emergency Management Glossary. Australian Emergency Manuals Series

- Frankel, D. S. (2003). Model Driven Architecture: Applying MDA to Enterprise Computing, John Wiley & Sons.
- Gharehdaghi, A. (2003). Design of a Generic Metamodel for Fieldwork Data Management. International Institute for Geo-Information, Science and Earth Observation Enschede, Netherlands. Master Theses.
- Krishna, A.; Clarke, R. J. and A. K. Ghose (2006) Genre-based approach to Requirements Elicitation Proceedings of the Eighteenth International Conference on Software Engineering and Knowledge Engineering (SEKE-2006), San Francisco, California, July 5-7 2006 ISBN1-891706-18-7, pp. 682-687
- Lagerstrom, R., P. Johnson, et al. (2010). Architecture analysis of enterprise systems modifiability - Models, analysis, and validation. Journal of Systems and Software 83(8): 1387-1403.
- Nygaard, K. (1988) Profession oriented languages in Andersen, P. B. and T. Bratteteig eds/ (1988) SYDPOL Program Working Group 2: *Computers and Language at Work: The relevance of languages and language use in development and use of computer systems* SYDPOL- Systems Development environments and Profession Oriented Languages, Institute of Informatics, University of Oslo, Norway Research Papers Series No. 126, ISBN 82-7368-030-4
- OMG (2002). Meta Object Facility (MOF) Specification, Object Management Group.
- OMG. (2003). MDA Guide Version 1.0.1 from <http://www.omg.org/docs/omg/03-06-01.pdf>.
- Opper, S, et al. (2009). "Timeline Modelling of Flood Evacuation Operations." Procedia Engineering, Volume 3, 2010, Pages 175-187
- Othman, S. H. and G. Beydoun (2010). Metamodelling Approach To Support Disaster Management Knowledge Sharing. Australasian Conference on Information Systems (ACIS'2010) Proceeding, Brisbane, Australia.
- Othman, S. H. and G. Beydoun (2011). A Disaster Management Metamodel (DMM) Evaluated. in Kang, B. and Richards, D. eds. PKAW 2010. LNCS (LNAI), vol. 6232, pp. 111-125, Springer.
- Othman, S.H. and G. Beydoun (2013), Model driven Disaster Management, Information & Management, Elsevier, 50 (5), 218-228
- Pfister, N., 'Community response to flood warnings: the case of an evacuation from Grafton, March 2001', in The Australian Journal of Emergency Management, Autumn 2002, Pp 19 – 29
- Seidewitz, E. (2003). "What Models Mean." IEEE Software 20(5).
- UNISDR (2009). UNISDR Terminology of Disaster Risk Reduction. Geneva, Switzerland.
- Vytautas Stukys, R. D., Aleksandras Targamadze (2010). "A Model-Driven View To Meta-Program Development Process." Information Technology and Control 39(2).