Expert Systems With Applications 63 (2016) 49–65

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A metamodel-based knowledge sharing system for disaster management

Siti Hajar Othman[a], Ghassan Beydoun[b,*]

[a] Department of Computer Science, Faculty of Computing, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia
[b] School of Computing and Information Technology, University of Wollongong, Wollongong, NSW 2522, Australia

ARTICLE INFO

Article history:
Received 16 February 2016
Revised 7 June 2016
Accepted 9 June 2016
Available online 25 June 2016

Keywords:
Knowledge sharing
Decision making
Disaster management
Metamodels

ABSTRACT

Timely knowledge sharing in disaster management (DM) is clearly vital, but it is remains challenging. Roles involved in DM processes often cut across many organizational boundaries and are dynamic. Knowledge involved is enormous and diverse. It includes information related to varieties of disasters, roles descriptions, plans and operations. Alas, practices may also vary across different regions and authorities. This paper makes a crucial contribution to address the knowledge sharing challenge by providing a knowledge based systems approach to facilitate structuring, storing and reusing DM knowledge.

The contribution of this paper is three folds: Firstly, it presents a metamodel-based architecture suitable for various distributed knowledge sharing settings; Secondly, it presents an actual implementation of such a system, the Disaster Management Knowledge Repository (DMKR1.0). DMKR facilitates collaboration and DM knowledge sharing using a tailored DM language. This offers a flexible structure to allow the storage and retrieval not only of observed and measured data, but also interpretative and inferred information of the disaster management knowledge. Thirdly, the paper provides disaster management exemplars of how DMKR users can easily instantiate DM models to communicate and to generalize their knowledge for the benefit of sharing it within their community. This presents a compelling evidence of the soundness and the effectiveness of the overall approach to DM knowledge sharing.

1. Introduction

Disaster Management (DM) processes involve many interacting elements e.g., people, authority, emergency teams, resources, procedures, uncertain environmental situations and many more. Modeling coordination of DM activities is tremendously hard and complex. DM activities often extend across various government sectors, non-governmental organizations/industry, from international levels down to state or region levels and may also include individual people and various facets of society. It is often unclear what are the exact tasks and responsibilities before, during or after a disaster strikes. The complexity of DM and the failures of DM agencies can be observed in many recent examples, e.g., the management of the Swine-Flu (H1N1) pandemic hitting Australian shores in large numbers through cruise ships in 2009 (Larcombe, Moloney & Schmidt, 2009), or the devastating communication failures in the Victorian bushfires in Victoria in 2008 (Cordner, Woodford & Bassed, 2011). Observed failures often surface as the required expertise not being available in a timely manner or a systemic inability to timely recognize and identify the required expertise. Reusing expertise is overlooked as it is often perceived as too specific to kinds of events such as floods, bushfires, tsunamis, pandemics or earthquakes.

This research advocates the use of a middle knowledge layer to enable DM practitioners to discern disaster dependent and disaster-independent features in the challenges that they face. For this purpose, a knowledge base system is developed. A DM generic metamodel is used in its development. The metamodel is called DMM and was developed recently in Othman, Beydoun and Sugumaran (2014). Thus we call the knowledge based system developed, Metamodel-based Disaster Management Knowledge Repository (DMKR). The structure of the repository is considered here an essential factor in obtaining good retrieval results as advocated in Shiva and Shala (2007). DMKR structure follows a tailored DM metamodel that DMKR facilitates knowledge reuse and timely identification of required DM knowledge sources. It allows a unified access to various DM experiences and offers a common DM

* Corresponding author. Fax: +61242214045.
E-mail addresses: hajar@utm.my (S.H. Othman), beydoun@uow.edu.au (G. Beydoun).

http://dx.doi.org/10.1016/j.eswa.2016.06.018
0957-4174/© 2016 Published by Elsevier Ltd.

description language. The work advocates that combining expertise used in various disasters will enable the best approach in managing a new disaster. It provides the following benefits:

- Facilitating communication among different disaster emergency users;
- Simplifying the teaching of new DM approaches through a set of semantic rules;
- Providing guidelines for creating DM models which can cover specific phases of DM (e.g., Response Phase *Earthquake Emergency Response Model*, Mitigation Phase *Bushfire Risk Reduction Model*);
- Highlighting scope for improvement in a given DM practice through validation against other existing DM models.

In an earlier work, we developed and validated a metamodel, to be used as language providing a middle layer of knowledge, to unify knowledge from different disaster experiences in Othman et al. (2014). This required tuning of a metamodelling process to the domain of DM to ensure that appropriate knowledge sources (candidate models) are identified and used as input to the process. The resulting metamodel, DMM, was theoretically validated showing how 89 DM models from the literature can be generated from the DMM (Othman et al., 2014). In this paper, we deploy DMM by demonstrating a viable and a unifying DM knowledge repository. This paper develops an actual knowledge sharing system which stores components for DM solutions and guides users on how to reconstruct solutions as new contexts arise. To store DM knowledge as reusable modelling components, our system, DMKR, has a metamodel driven interface to guide DM experts to articulate their knowledge appropriately. To illustrate the expressive power of the system, a number of case studies to describe real disaster situations are used in the DMKR implementation. Besides constructing DM solution models based for specific disaster problems presented to the system, DM stakeholders are also provided with the solutions from a range of disaster categories (e.g., earthquake, landslide, nuclear meltdown and etc.). This enables mixing and matching solutions across different disasters. Typical, users of DMKR can be disaster managers (local, state or federal), monitoring personnel, coordinators, aid agencies or even researchers who may wish to study the domain.

The rest of this paper is organized as follows: Section 2 describes the DM model driven background and related work that underpin the conceptual foundation of this paper. Section 3 describes the development and the architecture of a model driven DM system. Section 4 demonstrates the usefulness of the system implementing it and illustrating in real-world disaster problems. Specifically, a bushfire disaster exemplar is used to illustrate the knowledge capture and reuse facilitated by the system. Finally, Section 5 concludes with recommendations for future research.

2. Background and related work

A model is an abstract representation of a real world domain typically used to manage complexity (Beydoun & Hoffmann, 1998; Levendovszky, Rumpe, Schatz & Sprinkle, 2010). It generally consists of a collection of two elements: *concepts* and *relationships.* Concepts characterize domain entities and relationships characterize *links* between them (Trabelsi, Atitallah, Meftali, Dekeyser & Jemai, 2011). A model explicitly expresses the structure, behaviour and other properties in a domain and ideally has a causal connection to the real world (Aßmann, Zschaler & Wagner, 2006). Our approach for DM knowledge sharing can be said to be model-driven as advocated in (Othman & Beydoun, 2013). It is inspired by model driven systems development (Colin & Thomas, 2003; Stahl, Voelter & Czarnecki, 2005). Instead of requiring software developers to detail *how* a system is implemented, in a model-driven approach software models specify *what* the functionality and the architecture of the system to be used are (Colin & Thomas, 2003) via the use of software models developed in a specific language. With syntactically correct models, using only allowed symbols and conforming to rules, the modelling language facilitates sharing of the outcome of the modelling process. Developers are able to abstract and share knowledge using models as starting points. To formally describe the semantics underpining a formal modelling language, a metamodel is required, without which semantics of domain models can be ambiguous (2000).

Applying a model driven approach to share DM knowledge has the added benefit of also making knowledge accessible to non-technical specialists and newcomers to DM (Lauras, Truptil, & Bénaben, 2015; Xu, Wijesooriya, Wang & Beydoun, 2011). Similar to model-driven software development, this also requires a DM modelling language to specify all elements with which any model can be described. Model transformations can then enable precise and timely knowledge sharing communication across the various phases of the process (Trabelsi et al., 2011). Similarly, in DM, these are also required to enable the unified access to various DM knowledge models and to also communicate the knowledge across different disasters and DM activities.

The metamodel we use in this paper to specify DM modelling constructs is DMM. This is a DM specific metamodel which we synthesized and validated created in Othman et al. (2014). DMM has four sets of concept classes: the *Mitigation* (shown by Fig. 1), *Preparedness* (Fig. 2), *Response* (Fig. 3) and *Recovery* (Fig. 4) class of concepts. Each set represent a corresponding DM phase. This clearly describes the DM domain to its users. The relationship between the metamodel and domain model are described through *model transformations* (Vytautas Stuikys & Targamadze, 2010) which convert one model to another model (OMG, 2003). In this work, we deploy the transformations prescribed in the metamodelling framework of MOF, a standard for software metamodelling offered by OMG (2002). MOF defines a common way for capturing the diversity of modelling standards and interchange constructs that are used in model driven software engineering. It provides a framework for defining modelling languages or information models for metadata (Cook, 2004). It uses an *object-modelling* framework that is essentially a subset of the UML core. The main four modelling concepts in MOF are *classes, associations, data types* and *packages*. The main advantages of an OMG standard are its wide acceptance (Picka, 2004). MOF has four layers, M0, M1, M2 and M3. It has different views on modelling at different layers of details. It is strictly hierarchical. Concepts at any given layer (below M3) belong to a concept from the layer above. Any concept in any given layer (above M0) can be instantiated at the layer below. M3-Level is reserved for Meta-metamodel element, comprised of the description of the structure and semantics of meta-meta data. M2-Level is for the metamodel layer (instance of meta-metamodel), comprised of the descriptions of the structure and semantics of metadata. M1-Level is designed for the model layer (instance of metamodel), comprised of the metadata that describe data in the information layer. The lowest level, M0, is dedicated for user models (instance of model and also called as *information layer*). In this paper, M0-Level will cover the data that a disaster model describes at M1.

The system introduced in this paper, DMKR 1.0, represents how all the components associated the DMM are operationalised. It is inspired by method engineering, where a metamodel is used to index a software development methodology as required by a software process model (Firesmith, 2006; Tran, Beydoun & Low, 2007). The design of DMKR and the theories underlying it are described in the next section.