Metamodeling Approach for Hazard Management Systems

Anca Daniela Ionita[1] and Mariana Mocanu[2]
[1]*Department of Automation and Industrial Informatics, University Politehnica of Bucharest,*
Spl. Independentei 313, 060042, Bucharest, Romania
[2]*Department of Computing Science, University Politehnica of Bucharest,*
Spl. Independentei 313, 060042, Bucharest, Romania

Keywords: Metamodeling, Modeling Environments, Hazard Management.

Abstract: The management of natural and human-caused hazards is performed by reuniting a large variety of stakeholders, non-homogeneous collections of data, and systems that may not have been conceived for interoperability. The interdependency between hazards and the need of coordinated response also lead to the necessity to develop multi-hazard solutions, resulting in systems with a high complexity. This paper presents a metamodeling approach for hazard management systems, and a specific modeling environment, which considers the hazard, emergency, and geospatial views. The use of the model editor is exemplified on a system for early warning in case of accidental water pollution.

1 INTRODUCTION

Model Driven Engineering (MDE) may represent a solution for managing complex systems (Hossu et al., 2009) and coping with some of their critical properties, like size, heterogeneity, or the autonomy of their components (Bézivin et al., 2008). Furthermore, MDE was also found useful in case these complex systems are resulted from the integration of several legacy systems (Clavreul, Barais & Jézéquel, 2010).

This paper is focused on a kind of complex systems developed for prevention, early warning, and emergency action in case of natural or human-caused hazards. There have been several attempts to specify generic architectural frameworks for hazard management systems. Yet, there is no initiative of standardization that is independent of the type of hazard; the operating frameworks are preponderantly focused on a single hazard.

An important step towards unification within this domain was realized through the scientific reviews containing comparative analyses, according to criteria like: alarm levels, risk classes, event severity and likelihood, remaining time to hazard occurrence or arrival in the studied area (Villagrán de León, Pruessner, & Breedlove, 2013). There were also several attempts to specify frameworks generally appropriate for early warning systems, by defining guiding principles, stakeholders, preconditions, and strategies (UNDP, 2013). The United Nations adopted The Sendai Framework for Disaster Risk Reduction 2015-2030 (UNISDR, 2015) where priorities are specified at national and local levels.

At the technical level, diverse modeling approaches have been experienced, like the IDEF0 function modeling methodology and the EXPRESS language for data modeling (Fortier & Dokas, 2008). Variants for specific monitoring solutions like crowdsourcing also exist in the literature (Meissen & Fuchs-Kittowski, 2014). A very important progress has been made with the INSPIRE European directive regarding the spatial data infrastructure, meant to support interoperation at the level of data, metadata, monitoring and reporting (Bartha & Kocsis, 2011).

Still, there is no reference architecture for this domain, to increase the degree of reusability and to support the integration of existing systems.

The work presented here resulted in the definition of a metamodel and a modeling environment for the architecture of hazard management systems. Our research started with the identification of common and specific artefacts for hazard management systems, first for our research projects, then for other examples described in the publicly available documentation: systems in use, prototypes, conceptual frameworks etc. This led to the definition of a metamodel and of a modeling environment, used

Ionita, A. and Mocanu, M.
Metamodeling Approach for Hazard Management Systems.
DOI: 10.5220/0006417702570264
In *Proceedings of the 12th International Conference on Software Technologies (ICSOFT 2017)*, pages 257-264
ISBN: 978-989-758-262-2

for representing a variety of existing systems, and upgraded to support all the required links between artefacts, and to offer multiple views.

Section 2 presents the research method applied for metamodeling the hazard management domain. Section 3 explains the modeling paradigm – the metamodel and the hazard-specific modeling environment, which is applied for representing a water pollution early warning system.

2 RESEARCH METHOD

2.1 Study Hazard Management Domain

2.1.1 Study Documentation

The first step of our research, *Study Documentation* (see Figure 1), consisted of:

- Documenting the architecture of two hazard management systems developed in research projects we were involved to, the former for accidental river pollution (Ionita & Mocanu, 2015), and the latter for territorial vulnerabilities induced by nuclear facilities (N-WATCHDOG, 2017);
- Analyzing existing ontologies for hazards or vulnerabilities, like VuWiki (2016);
- Studying survey documents, including exhaustive classifications of hazards and a large set of examples that describe the current state of practice, e.g. (UNEP, 2012);
- Comparing the technology and the functionality through the examination of public documentation about systems in use for: disseminating alerts, e.g. Mobile Emergency Alert Systems (MEAS) (Park, Choi & Seo, 2014), and for early warning, e.g. (Kaku & Held, 2013);
- Studying scientific papers that propose new solutions and / or integratory approaches for the domain; our search was focused on multi-hazard systems and generic frameworks, e.g. (Balis et al., 2011);
- Looking for standards that have to be respected when implementing this kind of systems, e.g. INSPIRE or Common Alerting Protocol (CAP) (OASIS, 2010).

2.1.2 Collection of Conceptual Models

For a selection of the systems mentioned above, the study included:

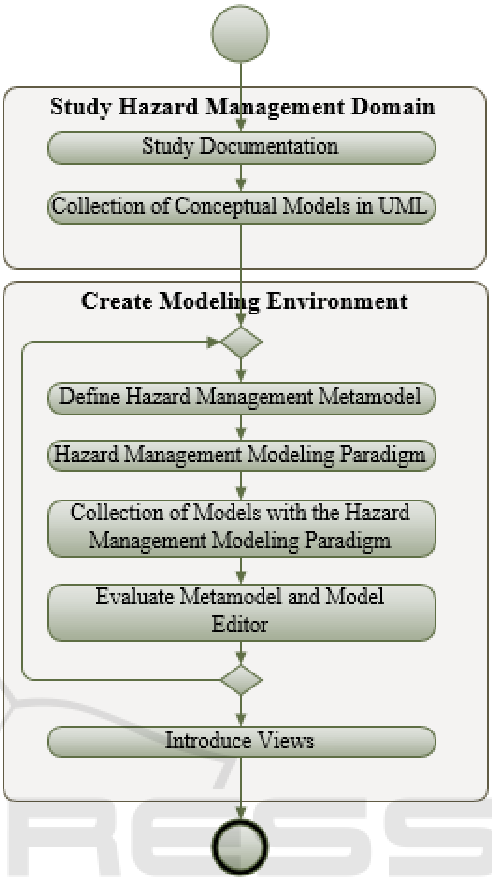


Figure 1: Research Method.

- Identification of the most important artifacts, i.e. modules, devices, stakeholders, concepts, with their relevant properties and functionality;
- Identification of dependencies between the artifacts mentioned above.

They were used to create a *Collection of Conceptual Models in UML* (Unified Modeling Language). UML was selected because it is a general-purpose modeling language, and it is object-oriented, similarly to the metamodel we were going to define. The collection currently contains about fifty models.

2.2 Create Modeling Environment

The steps for creating the modeling environment, presented below, were reiterated several times, for being capable to model all the studied systems.

2.2.1 Define Hazard Management Metamodel

Define Hazard Management Metamodel was the first step that succeeded the *Study of Hazard Management*

Domain. It started with a collaborative session, where several persons who studied examples of hazard management systems presented them to their colleagues. Each example was accompanied by a UML class diagram, showing the most important elements necessary to characterize that system. The goals of a collaborative session were:

- to identify the concepts that are common within the conceptual models;
- to make lists of terms with similar meaning;
- for each list, to find a term that abstracts the meaning of all the concepts within that list;
- to identify common properties of these concepts;
- to identify common relationships between them.

Afterwards, these elements were used for a first representation of the hazard management systems metamodel, based on the notation of UML class diagrams. This makes sense, as MOF (Meta Object Facility), the metamodel of UML, also uses the UML concrete syntax for its specification.

2.2.2 Hazard Management Modeling Paradigm

For creating modeling tools, we used Generic Modeling Environment (GME) (2017) and we built the *Hazard Management Modeling Paradigm*.

In the GME vocabulary, a paradigm is equivalent to a modeling language for the given application domain - in our case for hazard management systems. Thus, we formalized the abstract syntax, by mapping the UML metamodel (resulted at 2.2.1) to the metamodeling language supported by GME, i.e. compositions remained the same, associations became GME *Connections*, and classes became either *Atoms* or *Models*.

The concrete syntax of the language was also defined, by introducing icons specific to the new metamodel objects. They were used for configuring the model editor generated from the paradigm.

2.2.3 Collection of Models with the Hazard Management Modeling Paradigm

The editor was used for obtaining the *Collection of Models with the Hazard Management Modeling Paradigm*, i.e. all the models of systems initially represented with a general modeling language, UML, were transformed into GME models, conforming to the new paradigm / metamodel. In practice, they did not correspond to a single version of the paradigm, as it was upgraded over several iterations.

2.2.4 Evaluate Metamodel and Model Editor

The representation of models gave the opportunity to get to the next step: *Evaluate Metamodel and Model Editor*. The attempt to use the first version of a modeling language is not always successful; this is the moment to discover whether:

- a connection is missing;
- a modeling element has not been associated to an aspect and therefore it is not visible;
- two objects that should be connected belong to metamodel classes from different encapsulation levels;
- a part has not been included into a model;
- there are properties or concepts that cannot be instantiated from the existing metamodel, so new elements should be included.

Therefore, new versions of the modeling environment were necessary, and the steps for creating the modeling environment were re-iterated.

2.2.5 Introduce Views

The idea was to organize the architectural artefacts following the example of Enterprise Architecture (EA), considered as “Information Systems Architecture”. The reasons why a hazard management system can be addressed within the EA field are: the necessity to have a holistic approach, and the challenge of complex IT systems and organizational structures to meet business goals (Sessions, 2007).

As in hazard management, the fields of interest in EA cover more than software development, usually approached with view models (May, 2005). In the Zachman Framework classification (Zachman, 2016), the design artefacts are organized as a matrix, where the rows correspond to perspectives that represent the viewpoints of diverse stakeholders, including planner, owner, operator etc. Although Zachman talks about “views” and “aspects” without associating them with precise semantics, a perspective corresponds to an architectural viewpoint that governs an architectural view, composed of one or more architecture models, as defined by the ISO/IEC/IEEE 42010 standard (2011). Still within EA, the Treasury Enterprise Architecture Framework (TEAF) also included Functional, Information, Organizational and Infrastructure views (Urbaczewski & Mrdalj, 2006), which might be of interest for hazard management too.

Therefore, we looked for a framework that can depict the hazard management systems in a similar way, and group the metamodel abstract concepts. We identified three important views:

- *Hazard View* – embracing the concerns about the hazard itself, with its specific physical phenomena, theory, mathematical tools and techniques; this view should contain models for data collections, prediction, decision support and acquisition;
- *Emergency View* – regarding the viewpoint of the emergency professionals, who have to monitor the risk and take actions when necessary; it includes representations of the warning software and of the warning devices;
- *GIS View* - concerning geospatial distributions, collections of data, and visualization capabilities; for this view, one needs models for Geographical Information Systems (GIS) and Global Positioning Systems (GPS).

3 MODELING PARADIGM

3.1 Metamodel

The resulted metamodel contains a general part, for modeling the architecture inside a hazard management subsystem (as a graph of computing units connected through their interfaces), plus three parts that define specific models for the previously identified views, *Hazard*, *Emergency*, and *GIS* (see Section 2.2.5.) - and presented below.

We used the metamodeling language provided by Generic Modeling Environment, where an *Atom* is an indivisible modeling element, and a *Model* can contain other GME elements, to which it is connected through a line starting with a black diamond. An association that is not a containment is a *Connection*. The notation is similar to UML class diagrams.

3.1.1 Hazard View

The abstract syntax for the *Hazard* view, represented in Figure 2, considers that a hazard management system is composed of a model that characterizes the hazard, and of multiple models, correspondent to subsystems specific for managing the hazard, connected to each other.

Our study led to the identification of four kinds of models, all derived from *HazardManagementSubsystem*:

- *Data Collection*, corresponding to: assemblies of historical data on hazard events or demographic data, results of the vulnerable regions monitoring, scientific data, etc.;

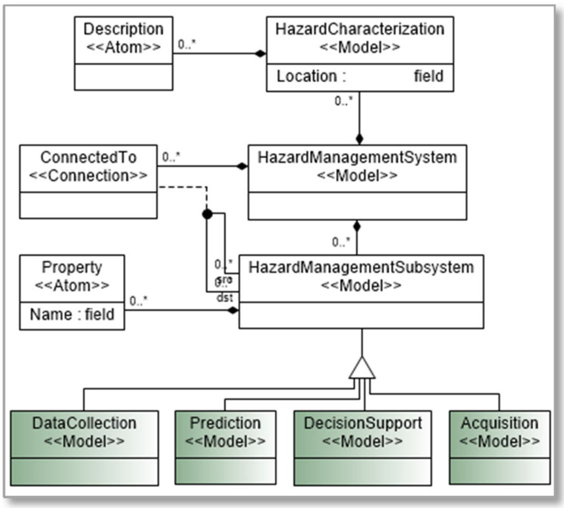


Figure 2: Metamodel for the Hazard View.

- *Prediction*, for subsystems that model the physical phenomena that drive the hazard and introduce predictive functionality;
- *Acquisition*, containing data acquisition subsystems, whose outputs are afterwards stored into *Data Collection* subsystems; they may be measuring devices, sensor networks, satellites etc. (Ionita & Olteanu, 2014);
- *Decision Support*, with units that analyze the available data and recommend predictive or response actions, which are transmitted to the subsystems pertaining to the *Emergency* view.

3.1.2 Emergency View

The abstract syntax for the *Emergency* view (see Figure 3) introduces two other kinds of models:

- *Warning Software* – meant for the efficient management of notifications sent to various actors and organizations, which represent potentially affected parts of emergency personnel; a *Notification* can be of type: *Information*, *Warning*, *Alert*, *Red Alert*, in respect with the severity of the hazard event;
- *Warning Device* – i.e. television set, radio set, telephone, computer, speaking-tube etc.

The organizations authorized to deal with emergency situations generally respect very strict processes; that is why the metamodel also contains an *Activity* model, with types that are specific to the emergency life cycle: *Preparedness, Response, Recover* and *Mitigation*, according to the classification adopted by the Federal Emergency Management Agency (FEMA) (Lindsay, 2012).

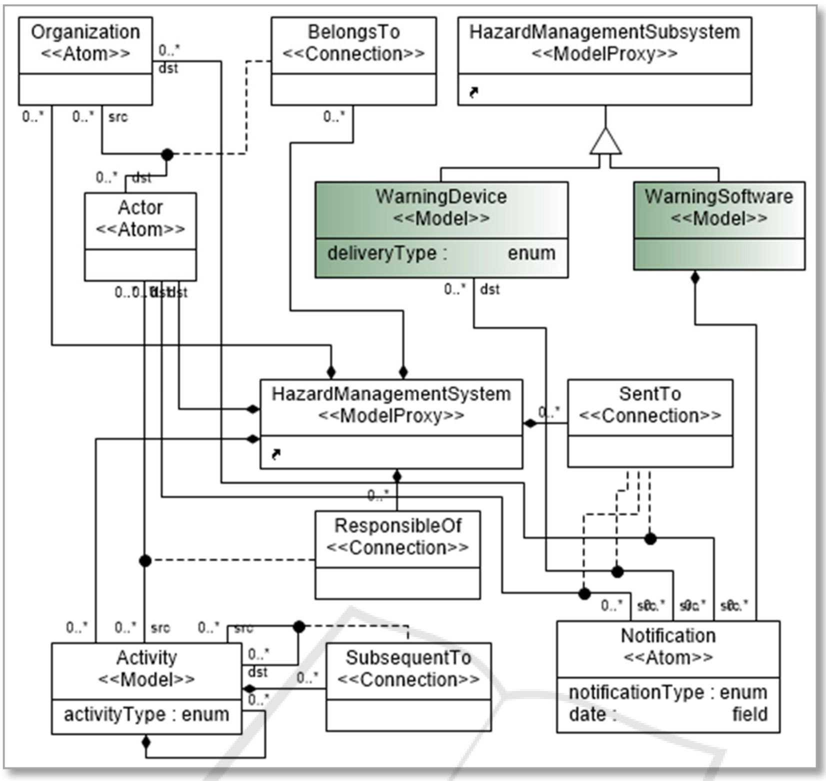


Figure 3: Metamodel for the Emergency View.

3.1.3 GIS View

For the *GIS* view, the metamodel introduces two models derived from *HazardManagementSubsystem*:

- *GeographicalInformationSystem*, which may contain *ComputingUnit* and *DataStore* objects, necessary for processing geospatial data and for representing geological, hydrological, or topological maps;
- *GlobalPositioningSystem*, for the physical system that provides geolocation data.

3.2 Hazard-Specific Environment

3.2.1 Modeling Editor

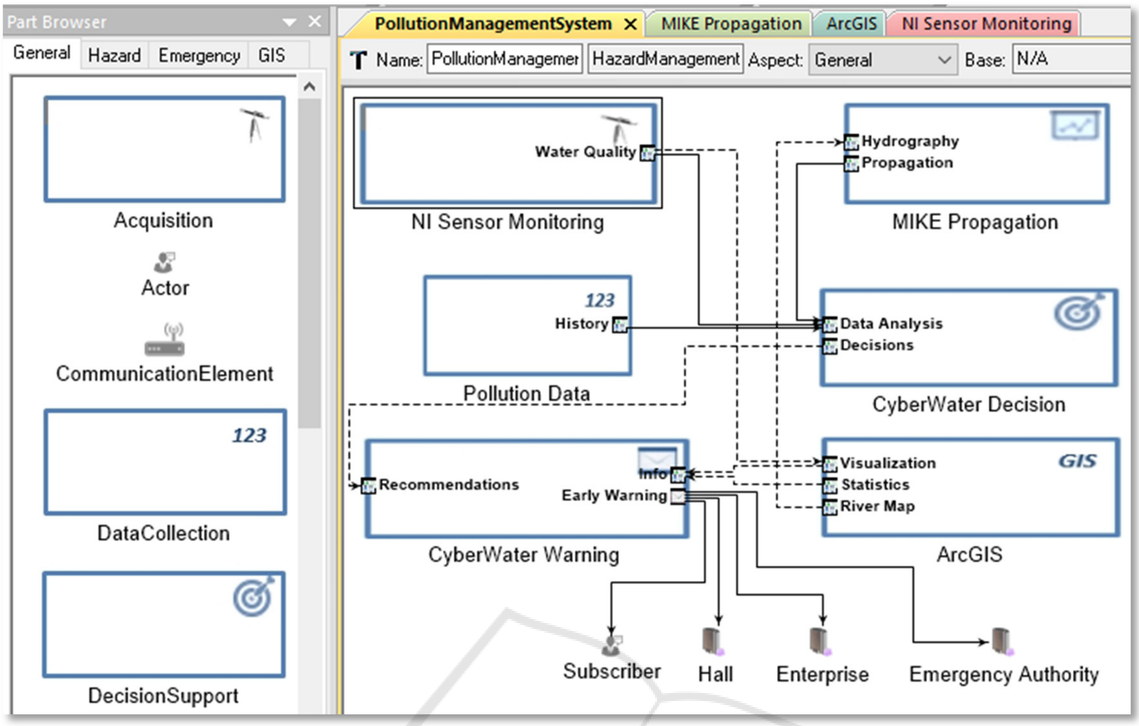
The metamodel, represented as a GME paradigm, was interpreted to generate a specific modeling editor for hazard management systems. For the configuration of its concrete syntax we introduced specific icons for all the metamodel elements described at 3.1. (see the Part Browser on the left side of Figure 4).

For each of the GME objects of type *Model*, like *Prediction*, *Acquisition* etc., the editor allows one to open a new tab with an editing pane, and represent a diagram for its internal structure, which is thus encapsulated; an exception was made for the objects of type *Interface*, which appear on the upper level, as ports. Thus, the editor supports several levels of encapsulation for describing a model.

For implementing the *Hazard*, *Emergency,* and *GIS* views, we introduced and configured three specific aspects; an *Aspect* is a GME concept used for controlling the modeling elements visibility. Thus, it is possible to draw the diagrams separately for each aspect (*Hazard*, *Emergency,* and *GIS*), to reduce the model complexity and allow a domain expert to see just the concepts for the correspondent viewpoint. Afterwards, if one gets to the *General* aspect, it is possible to see the entire model and to make connections between objects from different views.

3.2.2 Example of Model

Figure 4 presents a model edited with the hazard-specific environment, for a prototype that monitors the water quality of a river and recommends decisions to be taken in case one detects an accidental pollution (Ciolofan et al., 2013).



Name: PollutionManagemer HazardManagement Aspect: General Base: N/A

Figure 4: Pollution Management System Represented with the Hazard Management Editor.

The image presents the *General* aspect, but the model was realized as follows. Within the *Hazard* aspect, there are the following subsystems:

- *NI Sensor Monitoring* – a model of kind *Acquisition*, representing a wireless sensor network with a star topology, realized with National Instruments components, meant to collect surrogate data to be further processed for estimating the values for a set of physical quantities;
- *MIKE Propagation* – a model of kind *Prediction*, corresponding to a subsystem based on the MIKE environment (2017), capable to calibrate and execute the model for the propagation of a pollutant downstream; the purpose is to predict the moment a pollutant arrives at the localities downstream and what is its concentration; its inputs are real data on the shape of the river bed, and on the initial location of the pollutant detection;
- *CyberWater Decision* – a model of kind *DecissionSupport*, containing a rule engine and storing a collection of rules, activities, and threshold values; the decisions depend on the water quality detected by the sensor network, the predictions resulted from the simulations with MIKE, and the historical data;
- *Pollution Data* – a model of kind *DataCollection*, comprising data about previous pollution events in the area of interest.

Then, for the *Emergency* aspect, the metamodel contains:

- *CyberWater Warning* – a model of kind *WarningSoftware*, meant to transmit notifications towards private subscribers, organizations that represent local authorities, (e.g. village halls), industrial players (i.e. enterprises whose activity may be affected by the pollution) and, last but not least, to organizations that have the authority to take action in case of emergency situations;
- *Enterprise, Hall,* and *Emergency Authority* – atoms of kind *Organization*, receiving notifications from the warning subsystem;
- *Subscriber* – atom of kind *Actor*, representing a person that receives early warning notifications.

In the GIS aspect, we represented:

- *ArcGIS* – a *GeographicalInformationSystem* model, containing *ArcToolbox*, *ArcCatalog* and *ArcMap* (with the hydrographic basin maps) and exposing interfaces for visualization, statistics, and geospatial data related to the river.

Table 1: Summary of the objects of type *Model*, from the model represented in GME with the *HazardManagement* paradigm.

Aspect	Metamodel Level	Model Level		
	Kind	**Model**	**Port**	**Connected Port**
Hazard	Acquisition	NI Sensor Monitoring	Water Quality	Visualization
				Data Analysis
	Prediction	MIKE Propagation	Hydrography	River Map
			Propagation	Data Analysis
	DecisionSupport	CyberWater Decision	Data Analysis	Water Quality
				History
				Propagation
			Decisions	Recommendations
	DataCollection	Pollution Data	History	Data Analysis
Emergency	WarningSoftware	CyberWater Warning	Recommendations	Decisions
			Early Warning	Subscriber
				Hall / Enterprise
			Info	Statistics
				Visualization
GIS	GeographicalInformationSystem	ArcGIS	Visualization	Water Quality
				Info
			Statistics	Info
			River Map	Hydrography

Each of these subsystems is a model whose structure is represented inside and is not visible at the first level, except from the interfaces. Thus, in the *General* aspect we also connected interfaces of subsystems defined in different views. For instance, the *Hydrohraphy* interface from *MIKE_Propagation* - a model of type *Prediction*, which is part of the *Hazard* view - depends on the *RiverMap* interface of *ArcGIS* - a model from the GIS view. These connections are represented with dotted lines.

Table 1 summarizes the most important elements of this model, with their components that are visible as ports, and the ports from other models they are connected to. For each aspect, it presents the main objects of type *Model* (in the GME metamodeling language), with the metamodel entities they were instantiated from (i.e. the *Kind* in GME).

Note that the connections that transverse the views are easier to visualize in the GME diagram than in the tabular form. Moreover, one can select one of the three aspects, and visualize exclusively the parts of the model that correspond to them, thus obtaining a simplified diagram, and having the opportunity to reflect on further details that need to be represented.

4 CONCLUSIONS

For managing natural or human-caused hazards, there is a large variety of systems in place, and the trend is to introduce new infrastructure for monitoring the environment, and more efficient support for transmitting notifications when an undesired event happens. There is an increased interest to make exiting systems interoperate and to manage multiple types of hazards in an integrated way. This leads to the increase in complexity and non-homogeneity.

The paper proposed a metamodeling approach, which identified types of artifacts that are recurrent within hazard management systems, and used them for defining a metamodel and for configuring a specific modeling environment. The editor supports several levels of encapsulation in the representation of a hazard system, which is composed of various kinds of models that can be further described in separate diagrams, showing their inner parts and the connections between them. We also introduced the *Hazard*, *Emergency*, and *GIS* views, to reduce the complexity of modeling; one can represent diagrams correspondent to each view, and then visualize the elements situated at the top level of encapsulation, and introduce connections between them.

The approach has potential to be extended by composition with other metamodels and by adding model interpreters.

ACKNOWLEDGEMENTS

The work was realized within the Partnerships in Priority Areas Program - PN II, supported by MEN-UEFISCDI under the project number 298/2014.

REFERENCES

Balis, B, Kasztelnik, M, Bubak, M, Bartyński, T, Gubała, T, Nowakowski, P. & Broekhuijsen, J. (2011) The UrbanFlood Common Information Space for Early Warning Systems. Procedia Computer Science. 4. Elsevier, pp. 96-105. Available from: https://doi.org/10.1016/j.procs.2011.04.011 [Accessed 14th May 2017].

Bartha, G. & Kocsis, S. (2011) Standardization of Geographic Data: The European INSPIRE Directive. *European Journal of Geography*, 2, 79-89.

Bézivin, J., Paige, R.F., Aßmann, U., Rumpe, B. & Schmidt, D. (2008) Manifesto - Model Engineering for Complex Systems. In: *Dagstuhl Seminar Proceedings. Perspectives Workshop: Model Engineering of Complex Systems (MECS)*. Schloss Dagstuhl: Leibniz-Zentrum fuer Informatik, Germany. Available from: https://arxiv.org/abs/1409.6591 [Accessed 12th March 2017]

Ciolofan, S.N., Mocanu, M. & Ionita, A.D. (2013) Distributed Cyberinfrastructure for Decision Support in Risk Related Environments, In *12th International Symposium on Parallel and Distributed Computing*, Bucharest, IEEE. pp. 109-115.

Clavreul, M., Barais, O. & Jézéquel, JM. (2010) Integrating legacy systems with MDE, In: *32nd ACM/IEEE International Conference on Software Engineering*. Cape Town: IEEE, pp. 69-78.

Fortier, S.C. & Dokas, I.M. (2008) Setting the Specification Framework of an Early Warning System Using IDEF0 and Information Modeling. In: *5th International ISCRAM Conference Information Systems for Crisis Response and Management,* Washington, D.C.: ISCRAM, pp. 441-450.

GME (2017) *Generic Modeling Environment*. Available from: http://www.isis.vanderbilt.edu/projects/ gme/ [Accessed 10th March 2017].

Hossu, D., Humaila, H., Mocanu, S. & Saru, D. (2009) Complex networks to model the economic globalization process, In: *IFAC Proceedings Volumes*, 42(25): Elsevier, pp. 62-67.

Ionita, A.D. & Olteanu, A. (2014). Domain specific models, knowledge and tools to support multiple learning styles for engineering students, *Revue Roumaine des Sciences Techniques – Série Electrotechnique et Energétique*, 59 (4), 423-432.

Ionita, A.D. & Mocanu, M. (2015) Multiple Modeling Paradigms Applied for Accidental Pollution Management, *Environmental Engineering and Management Journal*, 14(9), 2051-2060.

ISO/IEC/IEEE (2011). *Systems and software engineering — Architecture description*. ISO/IEC/IEEE Standard 42010:2011.

Kaku, K. & Held, A. (2013) Sentinel Asia: A space-based disaster management support system in the Asia-Pacific region. *International Journal of Disaster Risk Reduction*, 6, pp. 1–17.

Lindsay, B.R. (2012) *Federal Emergency Management: A Brief Introduction*. Congressional Research Service. Available from: https://fas.org/sgp/crs/homesec/R42845.pdf [Accessed 18th March 2017].

May, N. (2005) A survey of software architecture viewpoint models, In: *Sixth Australasian Workshop on Software and System Architectures*. Brisbane, pp. 13-24.

Meissen, U. & Fuchs-Kittowski, F. (2014) Crowdsourcing in Early Warning Systems. In: *7th International Congress on Environmental Modelling and Software*. San Diego, California: Elsevier, pp. 326-332.

MIKE, (2017) *MIKE 11. River Modelling Unlimited*. [online] Available from: https://www.mikepoweredby dhi .com/products/mike-11 [Accessed 1st March 2017].

N-WATCHDOG Project. (2017) Early Warning and Decision Support Soft System for the Anticipative Assessment of the Fast Dynamics of Territorial Vulnerabilities Induced by Nuclear Facilities. Available from: http://proiecte.nipne.ro/pn2/155-proiecte.html [Accessed 14th May 2017]

OASIS, (2010) Common Alerting Protocol Version 1.2. Available from: http://docs.oasis-open.org/emergency/cap/ v1.2/CAP-v1.2-os.pdf [Accessed 18th March 2017].

Park, S.H., Choi, J.R. & Seo, J. (2014) A Pragmatic Media-Sharing Device for ATSC Mobile DTV Broadcasting. *International Journal of Advances in Soft Computing and Its Applications*, 6(3), 72-81.

Sessions, R. (2007) A Comparison of the Top Four Enterprise-Architecture Methodologies [online] Available from: https://msdn.microsoft.com/en-us/library/bb466232.aspx [Accessed 15th March 2017].

UNDP, (2013) *Early Warning Systems Framework*. Civil Defence Commission, Guyana, United Nations Development Programme, Available from http://www.undp. org [Accessed 2nd February 2017].

UNEP, (2012) *Early Warning Systems: A State of the Art Analysis and Future Directions*. Division of Early Warning and Assessment (DEWA), United Nations Environment Programme (UNEP), Nairobi.

UNISDR, (2015) *The Sendai Framework for Disaster Risk Reduction 2015-2030*. United Nations Office for Disaster Risk Reduction.

Urbaczewski, L. & Mrdalj, S. (2006) A Comparison of Enterprise Architecture Frameworks, *Issues in Information Systems,* VII (2), 18-23.

Villagrán de León, J. C., Pruessner, I., & Breedlove, H. (2013) Alert and Warning Frameworks in the Context of Early Warning Systems. A Comparative Review. *Intersections*. 12. Bonn: United Nations University Institute for Environment and Human Security.

VuWiki, (2017) Vulnerability Ontology. [online] Available from: http://www.vuwiki.org/index.php?title =Vulnerability_Ontology [Accessed 10th March 2017].

Zachman, J.A. (2016) The Framework for Enterprise Architecture: Background, Description and Utility, [online] Available from: https://www.zachman.com/resources/ea-articles-reference/327-the-framework-for-enterprise-architecture-background-description-and-utility-by-john-a-zachman [Accessed 2nd September 2016].