

Daniel Sadilek, Falko Theisselmann, and Guido Wachsmuth

Graduate school METRIK

Humboldt-Universität zu Berlin

Institute for Computer Science,

10099 Berlin, Germany

Advisor: Joachim Fischer, Doris Dransch, and Eckhard Holz

(sadilek|theissel|guwac@informatik.hu-berlin.de)

Challenges for Model-Driven Development of Self-Organising Disaster Management Information Systems

1 Introduction

Disaster management information systems (DMIS) can help mitigate damages caused by natural hazards. For example, as an early warning system [1], it should automatically generate and disseminate warnings to people at risk. In case of disaster, DMIS should support relief teams with up-to-date information, allowing for better coordination and more efficient help. In addition, DMIS could facilitate emergency communication. It should also serve as a data source for scientists. This could be the basis for further analyses allowing improvement of reactions to future events.

One problem with DMIS is that it may be partially destroyed during an event. This can lead to some functional loss resulting in less efficient disaster management. Integrating decentralised, self-organising, wireless sensor networks (WSN) is a promising approach to extend the capabilities of DMIS [2]. WSN are expected to hold up functionality in case of partial destruction and thus may lead to a better preparation for disasters and mitigation of damages.

Technologies for DMIS, WSN, and its integration are subject to ongoing research. Our goal is to apply a model-driven approach to the development of such technologies. In this paper, we do not treat the technologies itself. Instead, we focus on challenges for the model-driven approach.

2 Challenges

Various domain experts. Developing DMIS that integrate WSN is an inherently interdisciplinary task. Experts from the domains geology, computer science, electrical engineering, physics, and, of course, disaster management have to collaborate. Modelling techniques must address their different needs. *Geo-scientists*, for example, model real-world processes. These models are the foundation for data collection and processing, thus for early warning. Some problems related to this work are uncertainty of data and model results, data availability and quality, and adaptation of models to environmental circumstances. *Public authorities* are responsible for precaution of disasters. They produce contingency plans, distribute information to task forces and the public, and finance

disaster management measures. Problems related to this are coordination of (often non-governmental) task forces, reliability of information, time pressure, estimation of risk and vulnerability, and the need to evaluate costs and benefits. *Computer scientists* provide the technical infrastructure of DMIS. This includes the overall system architecture, system components such as network protocols and middleware services, and technologies and methods for development, deployment, and maintenance.

Heterogeneous models. In the development of a DMIS, several models are involved. Typically, *geophysical* issues like seismic wave propagation are modelled in mathematical terms, e.g. sets of differential equations. Such models are developed, maintained, and revised by geo-scientists. From an *administrative* perspective, organisational structures, responsibilities, exchange of information, cooperation, and policies for disaster management need to be captured in models. This includes spatial information supporting coordination. Such models might be developed by domain analysts or by the administration itself. To manage a concrete disaster, such models need to be comprehensible to task force agents and administrative staff. In *software* development for DMIS, several models emerge, e.g. concerning network architecture, system architecture, distributed data and algorithms, functional and nonfunctional requirements, application logic, and user integration. Even though these models are developed, maintained, and revised mainly by computer scientists, other experts might be involved. For example, a distributed algorithm detecting seismic waves should be developed by geo-scientists. All these kinds of models address particular domain experts, rely on other models, and thus need to be integrated into the overall system.

Simulation. Most models involved in the development process of a DMIS can not be evaluated as part of the real system (possibly containing thousands of nodes), because of missing sensor data, lack of ability to control model parameters (e.g. battery level), and the need to examine a model independently from the rest of the system. Furthermore, most models are too complex to evaluate them by static techniques such as model checking. Thus, simulation is needed for model parameter validation, calibration, sensitivity analysis, and evaluation of models against recorded data of former disasters. Models must reflect those needs and provide precise execution semantics. Since different models are needed for one simulation (e.g. simulation of a distributed algorithm depends on a network topology model), model coupling has to be taken into account.

3 Approach

We want to provide each domain expert with modelling means that are intuitive, concise, and semantically precise to allow for simulation. We believe, that a general purpose modelling language on its own (e.g. UML or SDL) is insufficient for a model-driven development of DMIS. Instead, we suppose a combination of several domain-specific languages (DSL) to fulfill these requirements.

We envision DSLs at each layer of a possible system architecture. A DSL will be interrelated to DSLs of the same layer and possibly to DSLs of other layers. In the middleware layer, for example, DSLs for hardware characteristics of sensor nodes and hardware-requirements of services can support a DSL for deployment specification.

For each DSL a metamodel has to be defined. Such a metamodel will form the basis for precise semantics descriptions enabling simulation and code generation. Furthermore, we want to instrument metamodels for automatic tool support and model coupling. Therefore, all metamodels have to instantiate the same meta-metamodel, for which we favour MOF 2.0 [3]. In our work, we will build on existing metamodelling research. Our aim is to contribute to powerful technologies and mature tools for development, maintenance, and revision of models in the context of DMIS.

References

- [1] United Nations: Global survey of early warning systems. Pre-print version released at the Third International Conference on Early Warning (2006)
- [2] SAFER Project: SAFER – Seismic eArly warning For EuRope. (2006)
- [3] Object Management Group: Meta Object Facility (MOF) 2.0 Core Specification. (2006)