MODEL-BASED DEVELOPMENT OF SELF-ORGANIZING EARTHQUAKE EARLY WARNING SYSTEMS

Joachim Fischer, Frank Kühnlenz, Klaus Ahrens, Ingmar Eveslage,
Humboldt-Universität zu Berlin

Corresponding Author: Joachim Fischer, Humboldt-Universität zu Berlin, Department of Computer Science
Unter den Linden 6, 10099 Berlin, Germany; fischer@informatik.hu-berlin.de

Abstract. A new approach for Earthquake Early Warning Systems (EEWS) is presented that uses wireless, self-organizing mesh sensor networks. To develop the prototype of such IT-infrastructures, we follow a model-driven system development paradigm. Structure and behaviour models of network topologies in specific geographic regions are coupled with wave signal analyzing algorithms, alarming protocols, convenient visualisations and earthquake data bases to form the basis for various simulation experiments ahead of system implementation and installation. The general objective of these studies is to test the functionality of an EEWS and to optimize it under the real-time, reliability and cost-depended requirements of potential end-users. For modelling a technology mix of SDL/ASN.1/UML/C++ is used to generate the code for different kind of simulators, and for the target platform (several node types). This approach is used for realizing a prototype-EEWS developed within the EU project SAFER (Seismic eArly warning For EuRope) in cooperation with the GeoForschungsZentrum Potsdam (GFZ). The first operational area of that EEWS is preparation for Istanbul in a region threatened by strong earthquakes. The presented paper focuses on our adopted and developed tool-based modelling and data base techniques used in that project, that are general and flexible enough for addressing similar prototyping use cases of self-organizing sensor-based IT-infrastructures.

1 Introduction

The concept of Self-Organizing Seismic Early Warning Information Networks (SOSEWIN) is being developed within the EU-project SAFER (Seismic eArly warning For EuRope) [1] in cooperation with the GeoForschungsZentrum Potsdam (GFZ). The work benefits from the Graduate School METRIK [2] on disaster management, supported by the DFG (German Research Society). It focuses on the adoption of METRIK-technologies concerning self-organizing, ad-hoc communication infrastructures and model-based software development for prototyping Earthquake Early Warning Systems (EEWS).

The SAFER project aims to fully exploit the possibilities offered by the real-time analysis of the signals coming from seismic networks for a wide range of actions, performed over time intervals of a few seconds to some tens of minutes. These actions include the shutting down of critical systems of lifelines and industrial processes, closing highways, railways, etc., the activation of control systems for the protection of crucial structures, as well as supporting the rapid response decisions that must be made by emergency management (continuously updated damage scenarios, aftershocks hazard etc.) [4] [5].

Present EEWS have a number of problems related to insufficient node density due to the high costs per node necessary for the purchasing, installation and maintenance of the usual more sophisticated seismological stations. However, such problems can be solved by using a low-cost, self-organizing, ad-hoc mesh sensor network that avoids more costly planned infrastructure. Such self-organizing communication networks were already successfully used within other application areas. One example is the Berlin RoofNet [3], which demonstrates the feasibility to build an autonomous wireless communication network in the city of Berlin at a moderate budget.

This paper demonstrates how the concept of such self-organizing mesh networks can be extended and adopted for the development of low-cost EEWS prototypes. As in Berlin RoofNet inexpensive Commercial-Off-The-Shelf (COTS) hardware is used with Linux as operating system and existing communication technologies, such as IEEE 802.11g WLAN, which operates in the unlicensed 2.4 GHz ISM band. Communication is close to real-time (delay ~ 0.5 - 1.0s), robust (mesh-structure with redundant paths) and based on the Internet Protocol, allowing for easy integration with existing applications and with the external public Internet (where available) [6]. Low-cost ground acceleration seismometer and GPS receiver are the basic components to recognize wave signals in dependence of time and locality.

A SOSEWIN network consists of nodes of different types with slightly different tasks. The elementary tasks are

- Routing Task: forwarding of received messages by wireless communication,
- Sensing Task: monitoring ground shaking using seismometer and GPS functionality,
- Alerting Task: issuing signals for alarms and reset alarms at different levels,

- Management Task: supporting installation, maintenance and control SOSEWIN for different manager types (seismological or network experts),
- Visualizing Task: supporting visualisation of SOSEWIN state information for different end users (public, decision maker in disaster's management).

In principle, each of the SOSEWIN nodes must undertake all of these tasks. However, there are different restrictions, depending upon the node's equipment and the task requirements. Fig. 1 shows a simplified SOSEWIN topology with typical nodes.

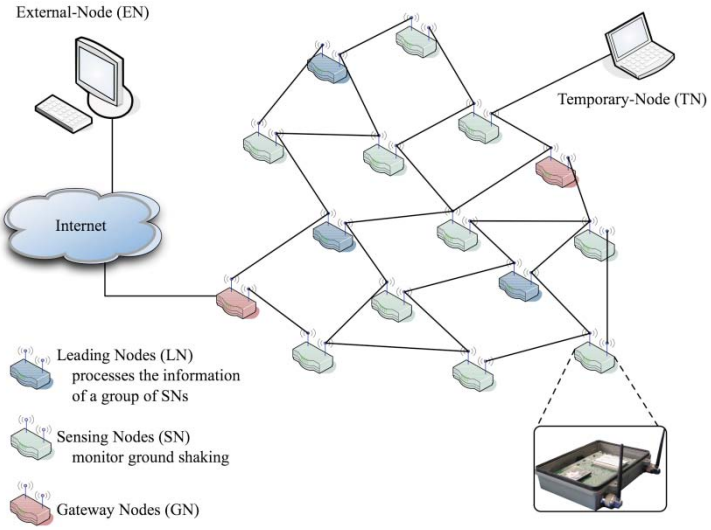

Fig. 1 A SOSEWIN example topology with typical nodes.

Developing the complex IT-infrastructure, we follow a model-driven system development paradigm. Structure and behaviour models of network topologies in specific geographic regions are coupled with wave signal analyzing algorithms, alarming protocols, convenient visualisations and earthquake data bases to form the basis for various simulation experiments ahead of system implementation and installation.

The general objective of these studies is to test the functionality of an EEWS and to optimize it under the real-time, reliability and cost-depended requirements of potential end-users. For modelling a technology mix of SDL/ASN.1/ UML/C++ is used to generate the code for different kind of simulators, and for the target on several nodes. This approach is used for realizing a prototype-EEWS developed within the EU project SAFER in co-operation with the GFZ. The first operational area of that EEWS is already installed in Istanbul, a region threatened by strong earthquakes. However, first SOSEWIN model tests were realized by using historical earthquake data, recognized by a centralized seismometer network in Taiwan and synthetic sensor data generated by a tool based on the work of Wang [7].

Our paper is structured into several sections. The next section 2 gives some background information to the application area and motivates the impact of earthquake wave signal analyzing approaches. Section 3 summarizes the current situation in the development of EEWS. Especially the advantages of self-organized systems are discussed here. Section 4 describes the general concepts of our SOSEWIN prototype, developed in the ongoing SAFER project. The principles of our Alarming Protocol (AP) are outlined in section 5. Section 6 discusses this infrastructure, especially modelling concepts, the SDL compiler and simulation components. The current status of the SOSEWIN development is given by section 7 including a short description of the testbed installation in Istanbul. The last section 8 summaries the results.

2 Earthquake Waves, Early Warning and Rapid Response

Earthquakes produce different types of seismic waves. These waves travel through the earth and provide an effective way to create an image of both sources and structures deep within the Earth. In addition their analysis is the foundation for different activities in a disaster's management process, so for earthquake classification, early warning and first response.

There are four types of seismic waves: P-waves and S-waves (called body waves), Rayleigh waves and Love waves (called surface waves). Body waves travel through the interior of the Earth. P-waves (primary waves) are longitudinal or compression waves, which brings the ground into alternately compressed and dilated movement in the direction of propagation. In solids, these waves generally travel almost twice as fast as S-waves (secondary waves) and can travel through any type of material. In air, these pressure waves take the form of sound waves, hence they travel at the speed of sound. Typical speeds are 330 m/s in air, 1450 m/s in water and about 5000 m/s

in granite[1]. When generated by an earthquake they are less destructive than the S-waves and surface waves that follow them. Surface waves remain below the Earth's surface. They can be much larger in amplitude than body waves, and can form the largest signals seen in earthquake seismograms. Seismograms are more strongly excited by surface waves particularly when the seismic source (hypo centre) is in close to the surface of the Earth.

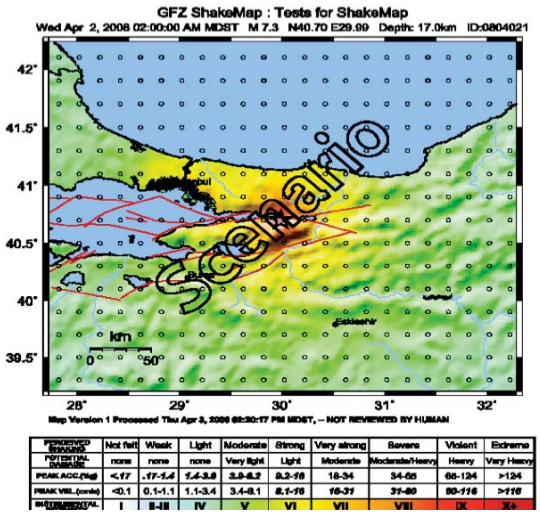

Fig. 2 ShakeMap example of a scenario earthquake.

It is not possible to predict an earthquake event. The only chance for preparation on the coming disaster is to use the most of the time delay between the arrival times of the P- and S-wave. In dependence of the distance between the epicentre of the earthquake (transferred hypocentre on the Earth's surface) and the critical area locations only few seconds to some tens of minutes remain for an early warning. The other important task in analyzing earthquake waves, which supports to save human life, is rapid response. This is a fast generation of so-called ShakeMaps, which show the wave peaks in the area (influenced by the earthquake event) in form of isobar lines or different colours. The combination of such ShakeMaps with existing building and inhabitation structures can offer start estimations of the disasters when these information would be available very fast after an event. A special kind of ShakeMap is an AlertMap. It is based on incomplete earthquake event descriptions (only on entrance signal data series) during the earthquake itself. The generation of such maps is an actual engineering challenge.

3 Earthquake Early Warning Systems

EEWS are based on the detection of P-waves that do not cause damage but precede the slower and destructive S-waves and surface waves. Dependent upon the distance between the hypocentre and the target area, a maximum early warning time before the S-wave arrives can be computed, based on wave travel characteristics and ground parameters. Therefore, the primary goal of an EEWS is simple: maximizing the early warning time under a minimal number of false alarms (which includes false positives and false negatives). An important secondary goal is to generate AlertMaps.

In the next subsection the existing approach for EEWS are described, followed by a statement about the problems these systems are faced. Subsection 3.3 illustrates how the vision of a decentralized sensor network can aim the disadvantages and lead to a better solution.

3.1 Present: Centralized Approach

Present EEWS always use a centralized approach (for example in the Marmara region, Turkey [9], Southern Apennines, Italy [10] and Taiwan [11]). Each station delivers its measured data or the alarm message for the case of P-wave detection over a (more or less) direct connection to a central data centre (which usually has a secondary data centre for backup). Within the data centre, it can then be decided whether an early warning message should be issued to the end users (e.g. nuclear power plants) who can then decide what actions will be instigated.

In the case of the already existing Istanbul Earthquake Rapid Response[2] and Early Warning System (IERREWS), ten strong-motion stations were placed as close as possible to the Great Marmara Fault zone, forming the online-sensor-part of the early warning system [9]. These stations are connected to the data centre of the Kandilli Observatory and Earthquake Research Institute via a digital spread spectrum radio link and continuously deliver ground-motion data for archiving and early warning purposes. Depending upon the location of the earthquake's epicentre and the recipient facility, the early warning time can be as high as about 8s [9].

3.2 Dilemma of current EEWS

Current early warning systems, like the above described IERREWS, often consist of only a few, but expensive (several thousands Euros) stations. This fact results in a number of problems:

Malfunction: If one station breaks down, then the area that it would normally observe can now only be monitored from afar, resulting in time delays that could seriously compromise the network's early warning capacity.

[1] Dependent upon the geology of the specific region and the hypocenter depth, P-waves travel at 5-8 km/s, and S-waves at 3-7 km/s.

[2] The Rapid Response part of the IERREWS comprises about 100 stations that report only peak ground accelerations every 20s in the event of an earthquake. This system is used mainly for rapid damage assessment after an earthquake.

Instrumental Density: This problem is related to the generation of precise information about an earthquake's intensity for city square cells, the size necessary being generally of the order of 500 m. Civil protection experts need such detailed information for reliable loss estimation maps (destroyed buildings, injured people and fatalities) that are the basis for effective planning by rescue teams. By comparison, EEWS usually have a station spacing of several kilometres.

Cost: However, increasing the density of seismic stations is limited by their expense.

Communication: The reliable transmission of all station information to central data stations or civil protection headquarters is very important, especially following an earthquake, where usually centralized communication infrastructures may have collapsed.

3.3 Vision: Decentralized Approach based on Low-cost Wireless Ad-hoc Mesh Sensor Networks

The basic idea presented in this work aims to avoid the problems identified above by deploying a much higher number of much cheaper stations (costing only a few hundreds of Euros per station, which is of the order of 10% compared to a classical station).

Another cost factors are the communication modules necessary for the link to the central data centres (in some cases within IERREWS, involving several hundred kilometres). Wireless, ad-hoc mesh sensor networks will allow much cheaper radio modules because a single station needs only to reach the nearest neighbour station, which would be only a few hundred meters away.

In addition, the reliability of such a mesh sensor net is a crucial point, since while single sensors may be destroyed, the whole system nonetheless can still detect the earthquake. This can be achieved because the sensor nodes act cooperatively in a self-organizing way. However, a number of challenging problems must be solved first (e.g. development of strategies for self-organization regarding the special requirements of EEWS; routing in huge multi-hop networks; deployment of software components).

The main advantages of such an approach, besides providing a more robust and cheaper architecture than centralized systems, may be summarized as follows:

- The simple deployment and installation of a temporary sensor net. This would be of particular value to, for example, groups such as the German Earthquake Taskforce, who deploy temporary arrays for the detection of aftershocks. Time consuming planning and (costly) installation of a traditional infrastructure-based system can thus be avoided.
- As mentioned above, in the event of an individual sensor node being destroyed, the self-organizing nature of the network will allow alternate communication routes to be established, while the information regarding the loss of the sensor or sensors may be utilized in damage assessments.
- At a later stage it is planned to provide the capability of using the network as an information system. In the event of a damaging earthquake, individuals will be able to send short messages such as "I am alive." or "I need help!" through a reliable mesh network that is still functional when other systems such as GSM may have collapsed.

4 SOSEWIN Overview

In contrast to existing EEWS, which are planned and centralized, we propose the use of a self-organizing ad-hoc wireless mesh network to overcome the problems of planning such a large network and administrating potentially thousands of Sensing Nodes (SNs). The advantages of such a network include robustness, independence of infrastructure, spontaneous extensibility as required, and a self-healing character in the event of failing SNs. However, these networks still pose a great research challenge, particularly regarding a routing-strategy to accomplish scalability requirements and time constraints.

To realise a hierarchical alarming system, each node runs the Alarming Protocol (AP) with different roles at runtime. The SOSEWIN nodes are organised into clusters using criteria that determines the optimum communications efficiency. Each cluster is headed by a SN that is designated, again based on communications efficiency, as a Leading Node (LN), with whom the other SNs within its cluster communicate general "housekeeping/status" information and initial alarms. The LN in turn communicated with other LNs, including the issuing of system alarms, based on each LN knowing the status of the nodes that make up their clusters.

For SOSEWIN, the following node types have been defined:

- *Sensing Nodes* (**SN**) monitor ground shaking. Most nodes in the network are of this type.
- *Leading Nodes* (**LN**) are basically Sensing Nodes as they consist of the same hardware. The "leading" property is a role that any SN can fulfil. A LN processes the information of a group of SN in its neighbourhood (usually not more than five SN).

- *Gateway Nodes* (**GN**) represent information sinks in the SOSEWIN that have connections to the end users (via the internet/satellite/cable) outside of the network, and are used for sending early warning messages. It includes the functionality of a SN.
- *External Nodes* (**EN**) are outside of the SOSEWIN and are connected via Gateway Nodes. They are to be informed first in the event of an alarm (e.g. GFZ, HU, Kandilli Observatory KOERI, police stations)
- *Temporary Nodes* (**TN**) are present in the network only for a short time to access data. An example of a temporary node is the laptop of an earthquake task force member, who wants to access ground shaking maps or waveform data.
- *Routing Nodes* (**RN**) ensure that communications between far-away nodes, which could not communicate otherwise. A Routing Node only delivers messages that it receives and undertakes no analysis. It is useful in being a low-cost way of extending the monitoring to a larger area.
- *Visualizing Node* (**VN**) A Laptop acting as a TN is a typical VN, which is able to come with a GUI to visualize subsequent SOSEWIN states on different abstraction levels by request. It's also easy to imagine that some of the SNs also have restricted visualization capabilities.

The AP uses peer-to-peer communication services realized by the underlying communication layer (TCP/UDP, OLSR [12], WLAN). To reduce the network load and the latency the messages are encoded using ASN.1 standard [14], [15] developed by ITU-T.

5 Alarming Protocol

The AP is one part of the SOSEWIN software architecture which specifies the algorithm and the rules how the sensor network detects an earthquake and issues the alarm to the end user. It consists of an ensemble of entities described in this section. Each entity is modelled as an extended finite state machine.

5.1 Overview of the system hierarchy

SOSEWIN supports a hierarchical alarming system. That's why the network is composed of node clusters, each cluster headed by a LN, where the cluster members are ordinary SNs or GNs. The definition of the clusters and the designation of which nodes are the LNs (which are themselves simply normal SNs within the network) are given by the initial installation. However, this can be changed dynamically if the network topology is changed. In principle, every sensor node can play two roles, a sensing and a leading role.

As Fig. 3 shows, the AP is realized by different asynchronous communicating protocol entities. Each entity is formally described by a state machine using the SDL-RT model description language:

- Signal Analyzing Entity **(SAE)**
  This one is responsible for analyzing the incoming streams of accelerometric raw data and informing the SE in the event of certain state changes. As presented in Fig. 3 we distinguish two different data sources for SAE. Alternately to a seismometer driver a stored data file (containing synthesized data) can be used here. This is controlled by data provider functions. An additional procedure is to save the raw data in a special data format (SEEDLink [13]) in a ring buffer.
- Sensing Entity **(SE)**
  It reacts on the results from the SAE trigger by informing its associated LE itself if the hosting node is a leading node (otherwise the LE of another node). To improve the SAE-SE-communication they operate over a semaphore-controlled common data base.
- Transport Entity **(TE)**
  It implements the message transport from the communications layer to the corresponding AP entities and vice versa. There are different message types (signal description, alarming, management …) with individual signatures. To do so it has the knowledge of the member node IP numbers of that cluster where it belongs to. A special task is the coding and decoding of all AP messages to and from other nodes. The coding/decoding procedure is realized by an ASN.1 compilation [14], [15]. For that, a C++ library modelling ASN.1 data types was used, developed by an earlier SDL/ASN.1-C++ compiler project [14].
- Leading Entity **(LE)**
  The LE monitors all associated SEs (that is the SE of the same node and the SEs of the Sensing Nodes within its cluster). An LE is able to cause or invoke group alerts and is also able to cause system alerts to be issued after communicating with other LEs; each system alert will sent to the TEs of all GNs. When a node is not a LN, then this entity is idle.

These entity ensembles (consisting of SAE, SE, LE, ME, and TE) have to be installed on the SOSEWIN SNs. In addition also the LEs have to be installed on SNs, but remain in an idle state. GNs have at least TEs.

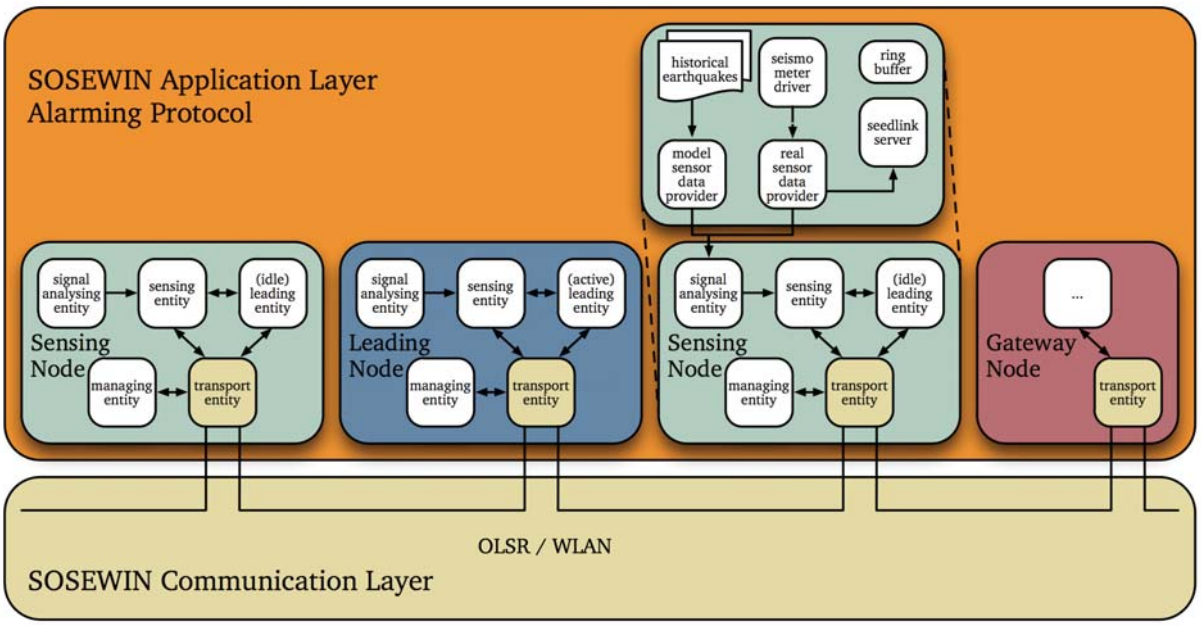

Fig. 3 Nodes and the associated protocol entities of the SOSEWIN Application Layer

5.2 Informal Protocol Description

Besides event detection, a general goal of the AP is to offer a network service that actualizes the knowledge about the states of all distributed SOSEWIN seismometers by their associated LEs as fast as possible after an individual change. The AP functionality is defined by two sub layers, an internal cluster protocol and a protocol for cluster-interaction. The internal cluster protocol defines the communications between a SAE and SE, and the communication between all SEs of a cluster and their representing LN. The inter-cluster protocol defines the communications between all LEs. If a critical number of P-wave triggers have reached the LE of a cluster's LN, this node informs its neighbouring LNs. In the case that a LE of a LN has received enough cluster alarms, a so-called system alarm will be sent as fast as possible to the GNs of SOSEWIN that are responsible for forwarding those alarms to defined ENs by peer-to-peer communication. According to this hierarchical principle, three alarm levels are recognized by the SOSEWIN:

- Pre-alarm (recognized by the LE of a LN, the requirement being a registration of a P-wave trigger by at least one SN of its cluster);
- Group alarm (recognized by the LE of a LN, the requirement being a certain number of node alarms of this cluster have been registered);
- System alarm (recognized by the LE of a LN, requires a certain number of group alarms registered by this LE).

Because of the independent reaction of the distributed nodes and their corresponding protocol entities, these alarm levels are reached by the individual nodes in a time-displaced manner. In addition to the three alarm states of the nodes represented by their protocol entities, two other states can be distinguished:

- Idle (recognized by the SE of each node, in that no event is occurring and preliminary analysis of the data input is going on);
- Final reporting (recognized by the LE of a LN, and the SE of all nodes, that the event is considered to be finished and the final data/result files (e.g. for ShakeMap) are being produced.

The AP is characterized by the principle that all SNs inform their LNs with as short time delays as possible about their current state without any explicit demand. Doing so, the LNs will be informed about the whole life cycle of an earthquake event according to their SNs. An explicit demand is necessary if ENs (via GNs) or TNs want to collect detailed information on the last event observed by the SOSEWIN. Once the first SN of a cluster has been triggered, the LN assigns an ID to the event, which will be based on the GPS time of the trigger at the first SN that detected it. The ID of the event is therefore the minimum event time, and maybe also with a code identifying the SN. Hence, both the real and false events will be recognized by the network by that code.

5.3 Formal Protocol Description

All of the protocol entities were described in detail by us in SDL, where in the first design stage the Real Time Developer Studio (RTDS) in version 3.4 was used, which supports an SDL dialect (SDL-RT [8]) in combination with UML class diagrams and C++ for activity and data type descriptions. This toolkit was extended in the context of the development of our EEWS prototyping infrastructure. Our extension allows not only a simulated execution and testing of the protocol entities, it also simplifies the code generation for an available cross com-

piler for the target hardware/operating system architecture. In addition to that the coding/decoding of messages for the node-to-node communication is realised by a developed ASN.1 compiler.

To ease the management of the complex prototyping task of EEWS an infrastructure was developed in parallel to the model development itself.

6 Prototyping Infrastructure

6.1 Infrastructure Components

The foundation of the tool integration in our EEWS prototyping infrastructure is a centralized management of models, software artefacts, and simulation results by several repositories that are implemented by data base technologies. Fig. 4 shows an overview on the core components realizing the identified requirements and concepts, which are shortly described in the following (from top to down in Fig. 4).

- *The Experiment Management System*
  supports planning, configuration, automated execution of simulations and storage of simulation results. It provides additionally GIS-based visualization capabilities for simulation results (e.g. Detection Maps) that can also be used for planning software deployment and monitoring of an installed SOSEWIN network.
- *The Model Repository*
  stores used SDL(-RT), UML and C++ models defining the entities of the AP. It also holds models of the environment (e.g. for network clustering, message transport properties or node breakdowns).
- *The Model Configurator*
  knows the target platform and uses platform dependent artefacts to configure the compiler (e.g. cross-compilation). It also specifies certain input parameters (e.g. threshold values or network clustering) and stores the whole configuration into the Experiment Repository.
- *The Transcompiler*
  is indeed a tool chain of several transcompilers, which accept SDL-RT models and compile C++ code at the end into different executable binaries (simulators, target code).
- *The Simulator*
  represents in fact a collection of several simulators of different functionality (derived from different simulation frameworks).
- *The Simulator Libraries*
  is a pool of simulation frameworks used by the simulators. They are used by the transcompiler to generate the executable binaries. Each simulator uses a different library with different features and restrictions.
- *The Earthquake Repository*
  comprises time series of historical recorded or synthetic generated earthquake data stored in a relational database system. Various input formats, such as (Mini-)SEED, SAC and several well-known-text (WKT) formats where mapped to the same database scheme, that is used as an uniform interface for simulations.
- *The C/C++ Libraries*
  providing the target code binary with threading and networking capabilities and with the necessary functions to decode and encode the network messages using the ASN.1 standard.
- *The Deployment Tool*
  enables the developer to distribute the AP in the testbed network from a remote site without access the nodes physically. Presently this is done by SFTP access from a central temporary node but a distributed approach is subject of actual research activities.

Using this infrastructure, different testbeds can be offered, namely for the detection of P-waves, for the functional correctness of different protocol concepts, and for the simulation of complete EEWS models. In addition, our infrastructure will be used for prototyping software components of the target EEWS. Additional a TN equipped by components of this infrastructure can play a temporary manager of the EEWS to visualize different dynamic installation, maintenance and operating activities.

Some of the concepts will described now more in detail.

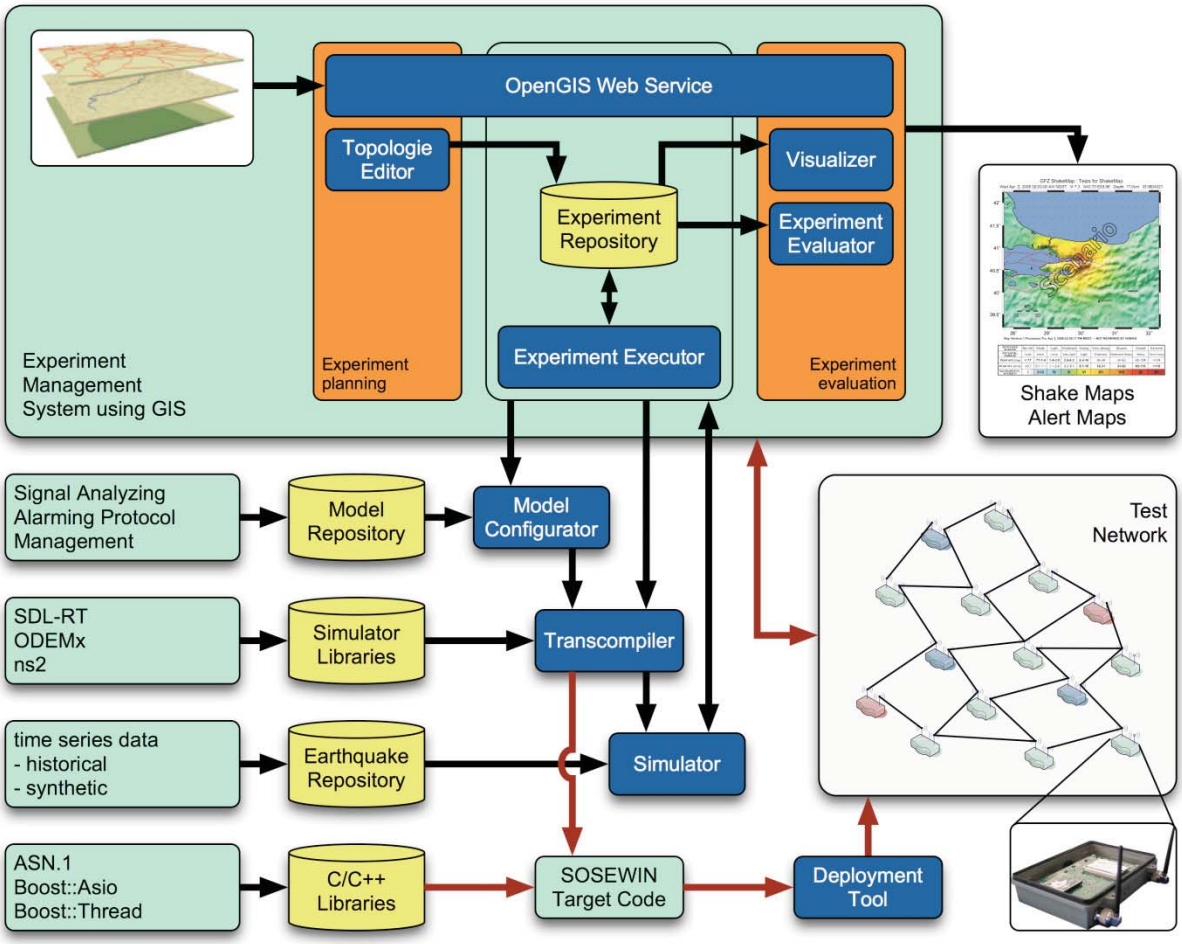

Fig. 4 Model-based Prototyping Infrastructure for EEWS.

6.2 Experiment Management System (EMS)

The results of the simulator runs (event traces) will also be stored within the relational database Experiment Repository (by import log files or direct access through an API), which is part of the Experiment Management System (EMS). Experimental results can then be evaluated manually by the Visualizer. This tool allows the presentation of a P-wave travelling through the network, with its detection (or non-detection) being marked by the sensor nodes changing colour (green to red, Detection Map). Other experimental output would include the so-called AlertMaps and ShakeMaps [12]. Both maps describe the spatial variation in the maximum ground shaking resulting from an earthquake for a given ground motion quality. A ShakeMap is generated from the complete time series of an event for each sensing node. In contrast, AlertMaps follow an evolutionary approach. Based only on the first few seconds of an earthquake's time series, a predicted ShakeMap is computed. Hence, while AlertMaps have a lower quality/accuracy than ShakeMaps, they are generated during an earthquake and are an early warning tool while ShakeMaps are used for post-event response planning.

For the configuration of EEWS models (network topology, software architecture of nodes, geographic area) under load (earthquake events, transmission disturbances) a graphical Topology Editor based on a Geographic Information System (GIS) is necessary. Adding and removing nodes is implemented using the OGC standard Web Feature Service (WFS). With WFS, a layer of spatial objects (e.g. points and lines with additional attributes) defined by the OGC standard Simple Features for SQL can be placed in a topographic map (overlay).

With our EMS, various automatic evaluations of the experiments can be computed. It considers the seismic wave velocities for a certain area and computes the estimated arrival of the P- and S-wave for each sensing node based on the hypocentre information in the repository. Then it checks the P-wave arrival time as determined by the sensing node, and determines whether this time is within a certain tolerance. Based on that mechanism, our EMS offers a comparison feature to evaluate different experimental results, for example the efficiency of different detection methodologies. Furthermore, it ensures reproducibility and consistency between the various development cycles of the simulator.

6.3 Transcompiler

By adopting PragmaDev-tools [23] our transcompiler follows the UML MDD approach to produce code for different platforms starting from SDL-RT model descriptions. This allows to process compositions of UML

(class, use case and sequence diagrams), SDL [22] (communicating processes) and C++ (data structures and sequential actions). We are able to use the RTDS[3] simulator to debug the model execution by SDL interpretation. In addition to this technique, other simulation frameworks can be coupled according to specific modelling and investigation requirements. Whereas by an SDL-based simulation, so far "only" functional characteristics have been examined, the ODEMx library ([17], [18], [19]) will allow non-functional performance characteristics of self-organizing systems to be determined by simulation, while varying the topology and environmental influences. The RTDS compiler is currently adopted by following extensions (shown in Fig. 5):

- annotations (prefixed SDL identifiers) in the SDL-RT source code allow a post-processing of the generated C-code, produced by RTDS,
- additional pattern-controlled transcompiler which transforms the generated C-code of RTDS to C++ supporting different targets. Using these patterns special parts of the structured RTDS C-Code will be substituted in each case following the related substitution patterns. Currently two alternatives are supported by our transcompiler:
    - transcompilation to ODEMx simulator, described in section 6.4, using the network simulator library ODEMx[4] which also handles time dependencies of state machine actions and message transportations by the network (as a main preposition for a model-based performance evaluation of SOSEWIN networks),
    - transcompilation by using Boost library thread and network functionality [20] to C++ as target code for the SOSEWIN nodes running a POSIX-compliant Linux, as they are currently installed in a testbed (see section 7).

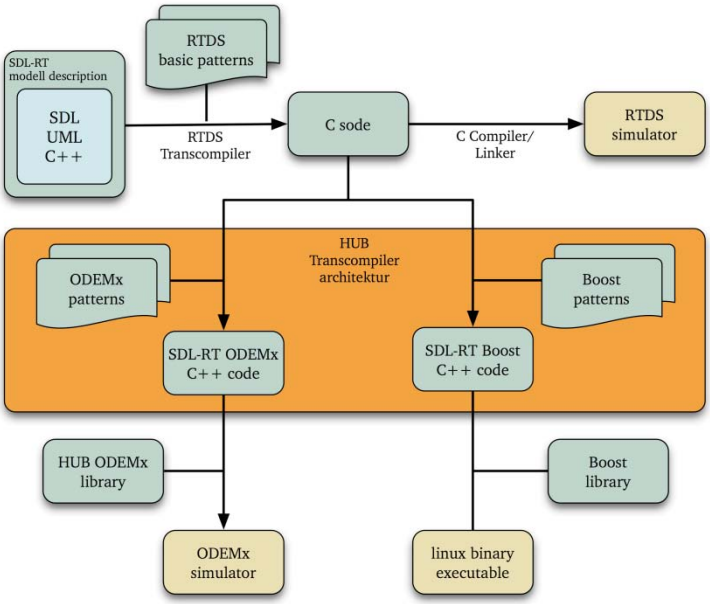

Fig. 5 SDL-UML-C++ Transcompiler Architecture.

6.4 Simulator

Our prototyping infrastructure integrates various simulators for different evaluation goals under common experiment management strategies. The last section has demonstrated different kinds of C/C++ code generations, where two of them were related with the simulation framework.

For the simulated execution of our formal described protocol entities we have to distinguish different main analyzing goals, where each of them is of certain complexity:

- functional evaluation of a single SAE SDL process by varying historical or synthesised earthquake data to check and improve the used signal analyzing numerical methods.
- functional evaluation of a single SN as an ensemble of different communicating state machines.
- functional evaluation of a configuration of SNs and LNs by varying historical or synthesized earth quake data,

[3] Real Time Developer Studio (V3.4) supports SDL-RT (a combination of UML, SDL, and C/C++) as a suited representation in the embedded / real time world today because it is basically a set of graphical representations of classical concepts such as tasks, messages, states, timers, and semaphores.

[4] Object-oriented Discrete Event Modelling is a C++ library for modelling and simulation of ensembles of discrete event driven processes combined with time-continuous processes.

- performance evaluation of a configuration of SNs and LNs by varying historical or synthesized earthquake data to estimate the capability of early warning.

A lot of parameters have to be tuned by the above scenarios in dependence on the target local area, the network size and topology, on the influence of environmental noise, on the behaviour of used underlying transport protocols. The successful realisation of this complex task to fulfil a compromise of different requirements is the base of continuation of further development steps, so for code generation, software deployment, network installation, network test, and finally for network operating.

All simulators produce MSCs and other event traces for a further information aggregation or visualisation. To simplify this kind of operations the trace raw data are managed by our experiment repository, realized as a data base system.

Simulator-I: Functional evaluation of signal analyzing algorithms of a single node. This simulator executes for a given number of SNs and provided time series of sensor's raw data (for each of them) the behaviour of their SAEs without any communication between themselves. With the help of this simulator an isolated test of the signal analyzing functionality can be realized. With the EMS topology editor the nodes can be positioned in a map. Using their GPS coordinates a synthesiser of an Earthquake can produce event data individually for each node by fixing a hypocentre and the earthquake parameters (e.g. rupture length, depth, energy). The simulator visualizes on one hand side the distribution of the earthquake waves in dependence of time and on the other side the P-wave detection by switching a virtual light controller from green to red by each of the node (Detection Map).

Simulator-II: Functional evaluation of alarming protocol. Here we use the RTDS SDL-RT-simulator to test the functional behaviour of smaller ensembles of the SOSEWIN nodes. We abstract from concrete earthquakes, and underlying protocol layers. One further important preposition is perfect transmission behaviour of used communication channels over the air. The results of functional tests allow us to evaluate and improve the logic of our alarming protocol. Typical outputs here are MSCs, which can be represented as XML and also stored in the experiment repository for further filtering by using data base functionalities.

Simulator-III / IV: Performance evaluation of alarming protocol in geographic context. Here we use the capability of our general-purpose ODEMx library [19], which supports the modelling and simulation of parallel process, where their state changes are described by discrete events in combinations with differential equations. This library contains especially concepts for simulation computer networks, where the protocol entities are extensions of the built-in ODEMx process concepts. Using this library two different simulators are produced by our transcompiler technology:

- Simulator III allows a integrated test of all developed software components in a large network of sensor nodes. Every node processes its own set of sensor data provides by a synthetic earthquake generator [7]. This simulator allows the estimation of required transmission times and transmission quality of alternative SOSEWIN configuration which guarantees the early warning functionality in dependence on different earthquake scenarios.
- Simulator IV should support in extension of SIMULATOR III the simulation of node breakdowns and of the behaviour of underlying protocol layers. Especially this simulator could be used for the training of disaster's management experts.

7 Current Status – A Prototype for Istanbul

Besides an existing small laboratory testbed of 10 SOSEWIN nodes at Humboldt-Universität, the main field test is installed in Istanbul. In order to establish a small network in the city of Istanbul, in 2008 a scientists group installed 20 nodes in the Ataköy area in the Bakirköy district (see Fig. 6) [24].

The testbed runs the SOSEWIN software architecture including the network communication layer and und application layer. The actual performance of the wireless network is satisfying and guaranties a reliable communication between the nodes. Two SOSEWIN nodes are connected with the internet, so the testbed is accessible from remote site.

The installation of the alarming protocol is planned in February 2009. In the test phase of the installed alarming protocol, the already mentioned generated synthetic sensor data will be used as input for the signal analysing algorithms. The experiments will give information about the performance of the early warning and rapid response capabilities.

Fig. 6 Actual SOSEWIN testbed installation in Istanbul.

8 Conclusion

We have presented a prototyping infrastructure for the model-driven development of EEWS based on self-organizing sensor networks. This architecture is based on OGC, OMG and ITU-T standards and combines different technologies for GIS, databases, behaviour modelling, code generation and simulation according to a special application domain by one integrated framework. So, it allows the evaluation of the real-time behaviour of projected earthquake monitoring and alarming systems and supports automatic code generation from evaluated structure and behavioural models. Modelling techniques which we used here are based on SDL and UML under special real-time requirements. Our prototyping infrastructure, implemented in C++, is used in the project SAFER for optimizing self-organizing seismic earthquake early-warning and rapid response systems, a real testbed is established in Istanbul.

An evaluation of the real-time behaviour of such complex systems is almost impossible or too expensive without prior modelling experiments, involving computer simulations. For that we identified several investigation goals supported by different simulators. This involves functional and performance evaluation of EEWS models by tuning topologies and parameters. Additionally to the model-based development our prototyping infrastructure supports also the installation, test, and operating of the network.

Currently the concepts of a cooperative signal analyzing are tested and the compiler technology reached a stable level. Now experiments have to evaluate and improve performance characteristics of the alarming protocol.

Although this contribution is naturally focussed on earthquake driven applications, the presented architecture of prototyping system may be adopted to those use cases where meshed sensor-based self-organizing infrastructures in combination with GIS are applied, such as in Heat Health Warning Systems [21].

9 Acknowledgement

The work was supported by the SAFER project partners at the GFZ J. Zschau, C. Milkereit, K. Fleming and M. Picozzi and by our colleagues from the Humboldt University J.-P. Redlich, B. Lichtblau and S. Heglmeier (Systems Architecture Group).

References

[1] http://www.saferproject.net, Seismic eArly warning For EuRope.

[2] http://www.gk-metrik.de, Model-based Development of Technologies for Self-organizing Systems.

[3] http://www.berlinroofnet.de.

[4] Wieland, M.: *Earthquake Alarm, Rapid Response, and Early Warning Systems: Low Cost Systems for Seismic Risk Reduction*. Electrowatt Engineering Ltd. Zurich, Switzerland 2001.

[5] Allen, R. M. and Kanamori, H.: *The Potential for Earthquake Early Warning in Southern California*. Science, 300, 786-789, 2003.

[6] Kurth, M., Zubow, A. and Redlich, J.P.: *Multi-channel link-level measurements in 802.11 mesh networks*. IWCMC '06, Canada 2006.

[7] Wang, R.: *A Simple Orthonormalization Method for Stable and Efficient Computation of Green's Functions*. Bulletin of the Seismological Society of America, 89:733–741, Juni 1999.

[8] http://www.sdl-rt.org, 2008/05/28.

[9] Erdik, M., Fahjan, Y., Ozel, O., Alcik, H., Mert, A. and Gul, M.: *Istanbul Earthquake Rapid Response and the Early Warning System*. Bulletin of Earthquake Engineering, 1(1):157–163, 2003.

[10] Weber, E. et al: *An Advanced Seismic Network in the Southern Apennines (Italy) for Seismicity Investigations and Experimentation with Earthquake Early Warning.* Seismological Research Letters, 78, 622-634, 2007.

[11] Lee, W. H. K., Shin, T. C. and Teng, T. L.: *Design and implementation of earthquake early warning systems in Taiwan.* Proc. 11th World Conference on Earthquake Engineering, Acapulco, Mexico, Oxford, England: Pergamon, Disc 4:2133, 1996.

[12] http://ietf.org/rfc/rfc3626.txt, Optimized Link State Routing Protocol (OLSR), RFC.

[13] http://www.iris.washington.edu/data/dmc-seedlink.htm, 2008/05/28.

[14] Fischer; J. et. al: *A Run Time Library for the Simulation of SDL'92 Specifications.* Proc. of the 6th SDL Forum.

[15] ITU-T X.690 *ASN.1 Encoding Rules: Specification of Basic Encoding Rules (BER).* International Telecommunication Union, 1992.

[16] Wald, D. J., Worden, B. C., Quitoriano, V. and Pankow, K. L.: *ShakeMap Manual; Technical Manual, Users Guide and Software Guide*. U.S. Geological Survey, version 1.0 6/19/06 edition, 06 2006.

[17] Fischer, J. and Ahrens, K.: *Objektorientierte Prozeßsimulation in C++.* Addison-Wesley Publishing Company, 1996.

[18] Gerstenberger, R.: ODEMx: *Neue Lösungen für die Realisierung von C++-Bibliotheken zur Prozesssimulation*. Diplomarbeit Humboldt-Universität zu Berlin, 2003.

[19] http://sourceforge.net/projects/odemx, 2008/05/28.

[20] http://www.boost.org, 2008/05/28.

[21] Endlicher, W., Jendritzky, G., Fischer, J. and Redlich, J.-P.: *Heat waves, urban climate and human health*. In: F. Kraas, W. Wuyi, and T. Krafft, editors, Global Change, Urbanization and Health, pages 103–114. China Meteorological Press, 2006.

[22] ITU-T Z.100: *Specification and Description Language (SDL).* International Telecommunication Union, 2002.

[23] http://www.pragmadev.com (RTDS V3.4), 2008/05/28.

[24] Milkereit, C., Fleming, K., Picozzi, M., Jäckel, K.H. and Hönig, M.: *Deliverable D4.4: Development of the seismological analysis software to be implemented in the Low Cost Network.* SAFER project report, 2008/05/09.