

Von den Anfängen einer objektorientierten Modellierung zur modellgetriebenen Entwicklung verteilter eingebetteter Systeme

JOACHIM FISCHER

fischer@informatik.hu-berlin.de

Am Beispiel eines prototypischen Erdbebenfrühwarnsystems für Istanbul als drahtlos kommunizierendes Maschennetzwerk von kleinen Rechnerknoten, ausgestattet mit preiswerter Sensorik, werden die Stärken eines modellgetriebenen Software-Entwicklungsansatzes gezeigt. Dabei kommen verschiedene objektorientierte Technologien zur Entwicklung verteilter eingebetteter Systeme zum Einsatz. Diese ermöglichen es insbesondere, dass der auf den Knoten zu verteilende ausführbare Code zur kooperativen Analyse seismischer Erschütterungswellen aus abstrakten Struktur- und Verhaltensmodellen in UML, SDL und ASN.1 automatisch generiert werden kann, die vorab verschiedensten modellbasierten Tests und simulativen Untersuchungen und damit iterativen Verbesserungen unterzogen worden sind. Dieser Beitrag ist meinen einstigen Lehrern an der Humboldt-Universität Roswitha März, Christoph Polze und Gunter Schwarze gewidmet.

1 Vorbemerkungen

Nach sechs Semestern Mathematik-Studium an der Humboldt-Universität, u. a. mit einer Ausbildung in Systemanalyse und Programmierung von Analogrechnern (bei Gunter Schwarze), sowie in der Programmierung von Digitalrechnern bei Einsatz von Algol-60 (bei Christoph Polze) und in den mathematischen Grundlagen numerischer Lösungsverfahren von Differentialgleichungen (bei Roswitha März), lernte ich 1976 an der Universität Warschau die Sprache Simula-67 kennen. Seit dieser Zeit begleiteten mich bis heute die Paradigmen *Klasse* und *Objekt*, *aktive* und *passive Objekt-Rolle*, *Ereignis* und *Prozess* sowie *Vererbung* und *Polymorphie*. Zunächst wurden diese Konzepte von der Fachwelt für exotisch gehalten und sowohl für die Modellierung als auch die Programmierung in der Bedeutung unterschätzt. Sie setzten sich erst später, insbesondere mit Smalltalk und C++, als universelle Programmierkonzepte durch und haben heute durch UML/SysML [1] die Systemmodellierung weltweit in voller Breite erfasst. Mehr noch, objektorientierte Metamodelle erobern zunehmend das Terrain der Definition von Computersprachen, insbesondere

von Sprachen mit graphischer Syntax, und lösen mehr und mehr grammatikalische Beschreibungen ab bzw. ergänzen diese.

Dieser Beitrag zeigt am Beispiel der Entwicklung eines prototypischen Erdbebenfrühwarnsystems für Istanbul als drahtlos kommunizierendes Maschennetzwerk von kleinen Rechnerknoten, ausgestattet mit preiswerter Sensorik, die Stärken eines modellgetriebenen Software-Entwicklungsansatzes. Dabei kommen verschiedene objektorientierte Technologien zur Entwicklung verteilter eingebetteter Systeme zum Einsatz. Diese ermöglichen es insbesondere, dass der auf den Knoten zu verteilende ausführbare Code einer kooperativen Analyse seismischer Erschütterungswellen aus abstrakten objektorientierten Struktur- und Verhaltensmodellen automatisch generiert werden kann, die vorab verschiedensten modellbasierten Tests und simulativen Untersuchungen und damit iterativen Verbesserungen unterzogen worden sind. Das Erdbebenfrühwarnsystem selbst ist das Resultat einer mehrjährigen interdisziplinären Zusammenarbeit des Instituts für Informatik der Humboldt-Universität zu Berlin mit dem Deutschen Geo-Forschungszentrum Potsdam [2].

2 Die Herausforderung

Erdbeben kündigen sich nicht an. Alle bisherigen Theorien zur präzisen zeitlichen Vorhersage von Erdbebenereignissen erwiesen sich als nicht haltbar. Bislang bleibt damit nur eine Alternative: Wenn die Erde, beginnend im jeweiligen Ereignisursprung (Hypozentrum) anfängt zu „wackeln“, könnten eventuell weiter entfernte Regionen vor den nahenden Bodenwellen gewarnt werden, wenn man sich die unterschiedlichen Laufzeiten der seismischen Wellen zu Nutze macht.

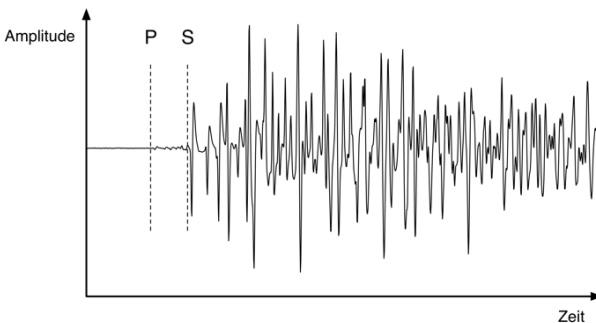


Abbildung 1: Von einem Seismometer erfasste Erdbeschleunigung einer P- und nachfolgender S-Welle

Interessanterweise sind dabei die schnellsten Wellen mit ~ 5 km/s (Primär- oder kurz P-Wellen) energieärmer als die langsameren zerstörerischen Sekundärwellen (S-Wellen) mit ~ 3 km/s. Primärwellen sind demnach als Ankündigungen der gefährlichen S-Wellen bestens geeignet (siehe Abbildung 1). Sie zu erkennen, um herannahende S-Wellen auch quantitativ zu prognostizieren, ist die zentrale Aufgabe eines Frühwarnsystems.

In der Praxis bleiben dafür (in Abhängigkeit der Entfernung vom Hypozentrum) nur wenige bis 100 s nach Erkennung der P-Wellenfront. Mit ihrer Installation lassen sich also lediglich die Auswirkungen von Erdbeben mildern. Nach einem schweren Beben kommt es somit immer darauf an, dass es Einsatzkräften schnellstens gelingt, ein hinreichend zuverlässiges Bild über die Lage (Zerstörungen, Tote, Verletzte usw.) zu erhalten, auch wenn die üblichen Kommunikationsinfrastrukturen zusammengebrochen sein sollten. Katastrophenexperten gehen davon aus, dass sie erste brauchbare quantitative Angaben durch eine Hochrechnung erhalten könnten, wenn denn die Spitzen der Erdbewegungen in einem Raster von 1 km für die betroffenen Regionen vorliegen würden.

3 SOSEWIN – Prototyp eines seismischen drahtlosen Maschennetzwerkes mit Sensorik

Das installierte System SOSEWIN ist nicht nur das weltweit erste drahtlose seismische Maschennetzwerk, sondern kann bei weiterem Ausbau das erste Erdbeben-Rapid-Response-System werden, das in einer dicht besiedelten Metropole mit Messtechnik vor Ort zum Einsatz kommt. Grundlage der Erdbebenfrühwarnung ist die bereits beschriebene Ausnutzung der unterschiedlichen Ausbreitungsgeschwindigkeiten seismischer Wellen, kombiniert mit einer kooperativen Signalanalyse, an der sich mehrere Seismometerknoten beteiligen.

Seit August 2008 läuft das System SOSEWIN in einer Konfiguration von 20 Knoten stabil und steht für verschiedene Studien zur Verfügung. Das Netz, im Stadtbezirk Ataköy auf Dächern 10-stöckiger Häuser mit Sichtkontakt installiert, ist per Internet mit dem Kandilli Observatorium in Istanbul, mit dem Deutschen GeoForschungsZentrum Potsdam und dem Institut für Informatik in Berlin verbunden (siehe Abbildung 2).

Seismometerbeobachtungen, inklusive permanent erfasster Seismometerrohdaten sowie beliebige Statusinformationen der einzelnen Knoten und ihrer Verbindungen lassen sich von den entfernten Standorten abfragen.

Ebenso wird (zumindest in bestimmtem Rahmen) ein entfernter Softwarekomponentenaustausch unterstützt. Die Einbindung des Systems in die bestehende Frühwarninfrastruktur vor Ort steht jedoch noch aus. Nach Bereitstellung eines Synthesizers für synthetische Erdbebendaten wurde des Weiteren begonnen, systematisch die Reaktion des Systems szenariorientiert zu

untersuchen. Weitere Erfahrungen mit der SOSEWIN-Technologie konnten durch die Anwendung unserer Netztechnik zur Gebäudeüberwachung im Zuge der intensiven Serie von Nachbeben des L'Aquila-Erdbebens in Italien (April 2009) durch die GFZ-TaskForce Erdbeben gesammelt werden [3].

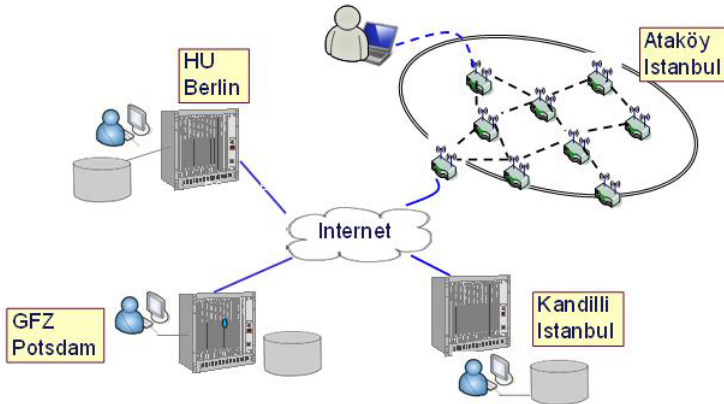


Abbildung 2: Erdbebenfrühwarn- und Rapid-Response-System für Istanbul

4 SOSEWIN – Technische Details

Die eingesetzte Hardware ist dabei mit kostengünstigen drahtlosen Routern vergleichbar, die heutzutage in vielen Haushalten zu finden sind.

Abbildung 3 gibt einen Überblick über die einzelnen Hardware-Komponenten. Die zentrale Komponente ist ein Embedded PC. In der ersten Version wurde hierfür das WRAP-Board (Wireless Router Applications Platform) der Firma PC Engines verwendet. Ausgestattet wurde das WRAP-Board ursprünglich mit einem AMD Geode Prozessor (x86er-Befehlssatz) mit einer Taktung von 266 MHz und 128 MB Arbeitsspeicher. Inzwischen kommen die leistungsfähigeren ALIX-Boards (500 MHz, 256 MB Arbeitsspeicher) derselben Firma zum Einsatz.

Hieran sind über die Mini-PCI-Schnittstelle zwei 802.11 WLAN-Karten angeschlossen. Die normale Konfiguration sieht den Betrieb des Maschennetzwerks über eine WLAN-Schnittstelle vor, während die zweite Karte Benutzern erlaubt, sich mit einem Knoten über seinen Laptop wie mit einem handelsüblichen Access Point zu verbinden, und so auf den Netzwerkknoten oder das gesamte Netzwerk zuzugreifen. Aber auch andere Konfigurationen sind möglich, beispielsweise können beide Karten den Verkehr des Maschennetzwerks weiterleiten falls mehr Bandbreite benötigt wird. Außerdem können

die Karten auch auf das 5 GHz-Band umgestellt werden, um Interferenzen mit anderen Netzwerken zu vermeiden.

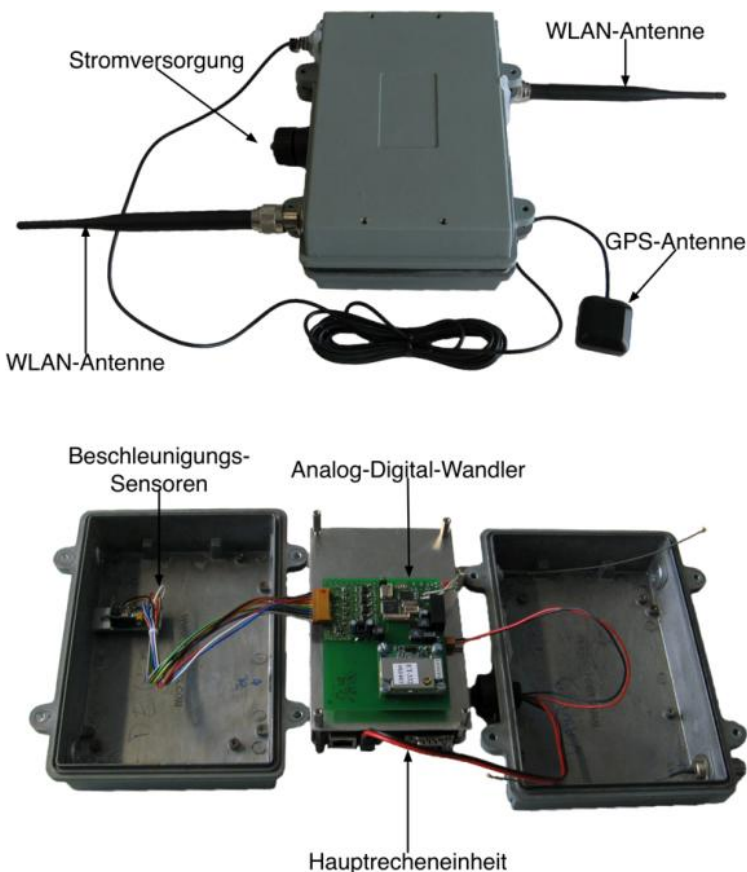


Abbildung 3: Ein SOSEWIN-Knoten

Eine CompactFlash-Karte dient als Massenspeicher für die Software und der Archivierung der anfallenden Messwerte in einem dafür angelegten Ring-speicher. Über die USB-Schnittstelle ist ein eigen entwickeltes Digitizer-Board angeschlossen, das drei Beschleunigungssensoren (MEMS-Accelerometers) die orthogonal in X-, Y- und Z-Richtung orientiert sind mit einem Analog-Digital-Wandler misst und mit GPS-Daten (Zeit und Ort) verknüpft. Diese werden dann per USB dem PC bereitgestellt. Das modulare Design des Digitizer-Boards erlaubt aber auch den Anschluss fast beliebiger anderer analoger Sensoren. Die gesamte Hardware findet in einem DIN-A5-großen und 4 cm dicken

wetterfesten Metallgehäuse Platz. Weiterhin existiert ein Ethernet-Anschluss, über den als Gateways fungierende Knoten mit dem Internet verbunden werden können, und so auf ein SOSEWIN-Netzwerk weltweit zugegriffen werden kann. Dies ist notwendig um das in Istanbul installierte Netzwerk administrieren oder in bestehende IT-Infrastrukturen integrieren zu können.

Die Software der SOSEWIN-Knoten besteht aus einer mehrschichtigen Software-Architektur (siehe Abbildung 4). Das Betriebssystem der SOSEWIN-Knoten, OpenWrt, ist eine spezielle für eingebettete Systeme entwickelte Linux-Distribution, die die üblichen und für die Funktion als Maschennetzwerk nötigen Softwarekomponenten integriert, wie z.B. das TCP/IP-Kommunikationsprotokoll und den SSH-Server-Dienst. Für die Wegewahl im Maschennetzwerk wird das selbstorganisierende Routingprotokoll OLSR (Optimized Link State Routing) [4] eingesetzt, dass sich in offenen Netzwerken wie dem Freifunk-Netzwerk in Berlin für hunderte von Knoten bewährt hat.

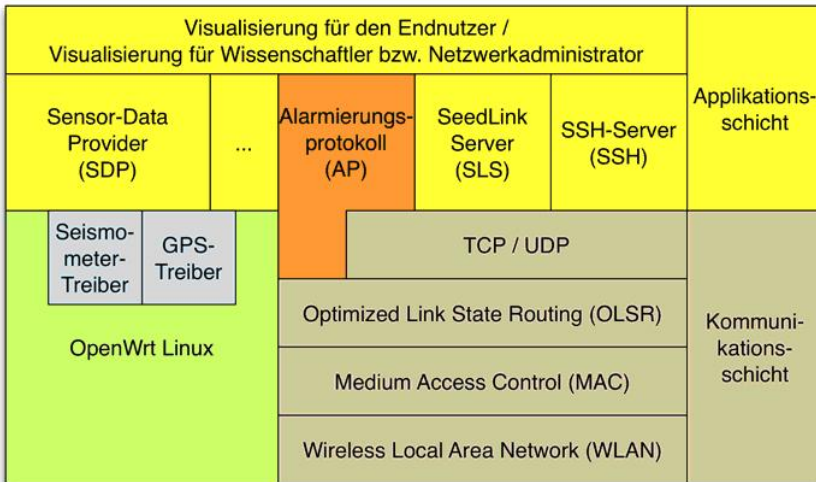


Abbildung 4: Die SOSEWIN-Software-Architektur

Auf Anwendungsebene sind die wichtigsten Komponenten der Seedlink-Server und das Alarming Protocol. Ersterer archiviert und stellt die anfallenden Rohdaten bereit, während das Alarmierungsprotokoll die Logik für eine kooperative Entscheidung über ein möglicherweise gerade stattfindendes Erdbeben und die Alarmierung in diesem Fall realisiert.

5 SOSEWIN – Software-Entwicklung

Bei der Entwicklung der für die Anwendung benötigten IT-Infrastruktur kam ein modellgetriebenes Entwicklungsparadigma zum Einsatz, das eine Konzeptüberprüfung und Fehleranalyse in der Design-Phase der Systementwicklung zum Ziel hat. Unterstützt wird dieses Paradigma durch ein bereitgestelltes GIS-basiertes Framework, welches in Form einer integrierten Umgebung Werkzeuge sowohl zur modellgetriebenen Entwicklung als auch zur Administration derartiger realer Netzwerke bereitstellt. Als eingesetzte Modellbeschreibungsmechanismen kommen bewährte und standardisierte Modellierungssprachen zum Einsatz, ähnliches gilt für die eingesetzten Basistechnologien.

Kern für die Integration der Werkzeuge ist ein Experiment-Managementsystem. Das Entwicklungsparadigma, die integrierte Werkzeugsammlung und das Experiment-Managementsystem werden im Folgenden genauer betrachtet.

Bei der Darstellung wird deutlich werden, dass die im Zusammenhang mit der SOSEWIN-Entwicklung entstandene werkzeuguunterstützte Methodik allgemeingültiger ist und insbesondere für die Einsatzvorbereitung unterschiedlicher Maschennetzwerke mit austauschbarer Sensorik, austauschbarer kooperativer Signalanalyse und austauschbarer Alarmierungssoftware o. ä. angepasst werden kann.

5.1 SOSEWIN-Paradigma einer modellgetriebenen Software-Entwicklung

Das SOSEWIN-Entwicklungsparadigma geht von der Bereitstellung von Struktur- und Verhaltensmodellen des zu erstellenden verteilten Systems zur Beobachtung und Bewertung seismischer Signale aus. Dabei werden die

- statische und dynamische Semantik des SOSEWIN-Systems beschrieben,
- simulative Verhaltensanalysen bei Variation bzw. Modifikation von
 - angenommenen Erdbebenereignissen,
 - Signalanalyseverfahren,
 - betroffenen geografischen Regionen,
 - aufgebauten Netztopologien,
 - alternativen Softwarearchitekturen und verwendeten Protokollen durchgeführt und
- letztendlich die automatische Codegenerierung für die spezifische SOSEWIN-Applikationssoftware realisiert.

Kern dieses Paradigmas ist also die geeignete Modelldarstellung, ihre Interpretation sowie die Transformation – und das alles selbstverständlich werkzeuguunterstützt.

Typisch für die Anwendung des Paradigmas ist die Durchführung einer Vielzahl systematischer simulativer Tests, die sowohl funktionale als auch nichtfunktionale Eigenschaften untersuchen und bewerten bevor der Zielcode erzeugt wird [5]. Ohne die Bereitstellung eines geeigneten Experiment-Managementsystems mit Datenbankbindung wäre auch diese Aufgabe nicht zu bewältigen gewesen.

Von besonderem Gewicht ist die Tatsache, dass die validierte Verhaltensmodellierung drahtlos kommunizierender Systeme gegenwärtig noch einen Forschungsbedarf aufweist, wo Fortschritte ohne Vergleiche von realem und modelliertem Verhalten undenkbar wären. Die Nutzung des für SOSEWIN entwickelten Experiment-Managementsystems zur Erfassung gelogger Ereignisse im realen Netzwerk (in Istanbul oder anderswo) und des Vergleiches mit Simulationsergebnissen ist damit ein weiterer wichtiger Aspekt des SOSEWIN-Entwicklungsparadigmas.

Da es sowohl für den Test des realen Systems als auch seiner Simulationsmodelle wiederum bereitgestellter Datensätze verschiedener angenommener Erdbebenereignisse bedarf, die für reale oder gedachte geographische Knotenpositionen vorliegen müssen, ist die synthetisierte Rohdatenbereitstellung eine weitere Anforderung für die Umsetzung des SOSEWIN-Entwicklungsparadigmas.

5.2 Architektur der SOSEWIN-Entwicklungs- und Administrationsumgebung

Die Entwicklung eines selbstorganisierenden Erdbeben-Frühwarn- und Rapid-Response-Systems mit Signalanalyse, verteiltem Alarmierungsprotokoll und Routing-Protokollen stellt eine komplexe Aufgabe dar. Die Umsetzung des SOSEWIN-Paradigmas geht davon aus, dass der Implementation und Installation auf der Zielplattform umfangreiche Modelluntersuchungen vorausgehen. Dabei kommen verschiedene Beschreibungssprachen für die unterschiedlichen Modelle zur Beschreibung von Funktionalität, nichtfunktionaler Leistungsparameter und Umgebungseinflüssen zum Einsatz. Die Untersuchung der durch die Modelle beschriebenen Systeme auf Leistungsfähigkeit und Korrektheit erfolgt jeweils durch Simulation der dynamischen Verhaltensmodelle. Die Erstellung von Simulatoren aus den Modellen ist ein mehrstufiger Transformationsprozess, wobei Transcompiler-Technologien zur Abbildung unterschiedlicher Modellierungssprachen nach C++ zum Einsatz kommen. Danach erfolgt die Kompilation der erzeugten C++-Dateien und das Linken mit Bibliotheken der in C++ implementierten Simulationswerkzeuge, woraus ausführbare Simulatoren entstehen. Diese können dann unter Verwendung einer geeigneten Last für die Analyse verwendet werden.

Die Editoren zur Erstellung unterschiedlicher Modelle, die Repositories zur Speicherung dieser Modelle, die Kompilations- und Transformationswerkzeuge, der Synthesizer von Erdbebendaten, die Simulationsbibliotheken und die jeweils erzeugten Simulatoren Abbildungen zusammen mit dem Experiment-Managementssystem die SOSEWIN-Entwicklungs- und Administrationsumgebung.

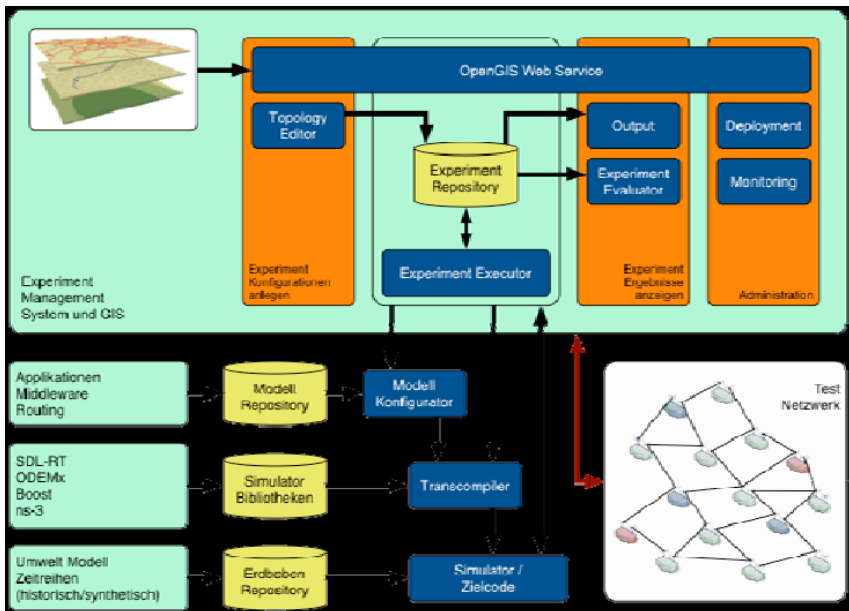


Abbildung 5: Architektur der SOSEWIN-Entwicklungs- und Administrationsumgebung

Abbildung 5 zeigt überblicksmäßig die Architektur dieser Umgebung.

- Das Experiment-Managementsystem ist die zentrale Komponente. Es unterstützt mit einem Topologie-Editor die Planung, Konfiguration und automatische Ausführung von Experimenten unter Verwendung von Modellen und Simulatoren eines Sensornetzes. Darüber hinaus wird auch der experimentelle Test des realen Systems bei vorherigem automatischen Softwarekomponentenaustausch und entfernter Umstellung der Sensordatenerfassung auf die Entgegennahme von Umgebungsdaten aus eingespielten Dateien unterstützt. Zusätzlich wird die Speicherung, Auswertung und Visualisierung der Ergebnisse unterstützt.

- Der graphische Topologie-Editor basiert auf einem Geographischen Informationssystem (GIS), welches wiederum auf der OpenGIS Web Service Architektur des Open Geospatial Consortium (OGC) aufsetzt. So lassen sich beispielsweise leicht mit dem OGC-Standard [6] Web Feature Service (WFS) Knoten in eine bestehende Netztopologie einfügen, ändern und entfernen.
- Die vom Managementsystem benutzten Modellrepositories dienen der Verwaltung von Zustandsautomatenmodellen der Anwendungsprotokolle für die kooperative Signalanalyse und Alarmierung in unterschiedlichen Stufen (Knotenalarm, Gruppenalarm, Systemalarm). Unterstützt werden Modelle in den Sprachen SDL(-RT), UML, ASN.1 und C++ für die Verhaltensdarstellung von Softwarekomponenten der Anwendung, Middleware- oder Routing-Protokollebene [7].

Der Modell-Konfigurator kennt die Zielplattform von SOSEWIN, kann dementsprechend plattformspezifische Software-Artefakte einsetzen um den finalen Cross-Compilationsprozess vorzubereiten. Weiterhin legt er spezifische Eingabeparameter, wie z. B. Schwellwerte für das P-Wellen-Detektionsverfahren und für die Auslösung unterschiedlicher Alarmierungsstufen fest und speichert die Gesamtheit der Konfigurationseinstellung.

6 Zusammenfassung und Ausblick

Mit SOSEWIN als Prototyp eines mit Sensorik ausgestatteten, drahtlos kommunizierenden Maschennetzwerkes ist es insbesondere gelungen, den zentralen Anspruch zu untermauern, dass selbstorganisierende Systeme in Gestalt derartiger Netztypen durch die Miniaturisierung ihrer Knoten, durch autonome Adaptionsfähigkeit und durch niedrige Kosten völlig neue Möglichkeiten eröffnen, komplexe Umweltprozesse zu messen und in Echtzeit kooperativ zu analysieren.

Das für SOSEWIN genutzte modellgetriebene und über eine integrierte Werkzeugsammlung unterstützte Entwicklungsparadigma erwies sich als äußerst qualitätsverbessernd und zeit- und ressourcensparend und kann für die Entwicklung anderer flächendeckender Umweltprozessanalyseverfahren angepasst werden.

Die Entwicklung von SOSEWIN profitierte von der Entwicklung von Basistechnologien des interdisziplinären DFG-Graduiertenkollegs METRIK, das ab Oktober 2010 in einer zweiten Förderphase fortgesetzt wird [8]. Ab September 2010 wird über einen vom BMBF geförderten Technologietransfer in die Wirtschaft, die Serienproduktion von SOSEWIN-Knoten ermöglicht, zunächst für Forschungszwecke, später für den realen Einsatz, um tatsächlich große, flächendeckende Netzwerke studieren und einsetzen zu können.

7 Literatur und Internetquellen

- [1] LANGUAGE, OMG: Unified Modeling. <http://www.omg.org/spec/UML>.
- [2] FISCHER, J.; REDLICH, J.-P.; ZSCHAU, J.; MILKEREIT, C.; PICOZZI, M.; FLEMING, K.; LICHTBLAU, B.; KÜHNLENZ, F.; EVESLAGE, I. & BRUMBULLI, M. (2010): Ein Erdbebenfrühwarnsystem für Istanbul. *Proceedings VDE Kongress, Leipzig*.
- [3] PICOZZI, M. ET AL. (2009): Real time monitoring of structures in task force missions: the example of the Mw=6.3 Central Italy Earthquake. *Natural Hazards*, 52 (2), S. 253-256.
- [4] OLSR, RFC 3626. <http://tools.ietf.org/html/rfc3626>.
- [5] FISCHER, J.; KÜHNLENZ, F. & AHRENS, K. (2008): Model-based Development of Self-organizing Earthquake Early Warning System. *In Joint ITU-T and SDL Forum Society workshop on „ITU System Design Languages“*.
- [6] OGC, Open Geospatial Consortium. <http://www.opengeospatial.org>.
- [7] SDL-RT. <http://www.sdl-rt.org>.
- [8] Model-based Development of Technologies for Self-Organizing Decentralized Information Systems in Disaster Management. <http://www.gk-metrik.de>.

Die zitierten Internetquellen wurden zuletzt am 30.08.2010 aufgerufen.