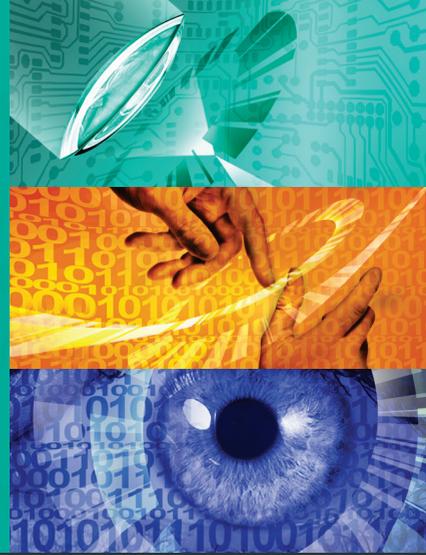


Karlsruher Schriften
zur Anthropomatik

Band 29



Jürgen Moßgraber

**Ein Rahmenwerk für die Architektur
von Frühwarnsystemen**

Jürgen Moßgraber

**Ein Rahmenwerk für die Architektur
von Frühwarnsystemen**

Karlsruher Schriften zur Anthropomatik

Band 29

Herausgeber: Prof. Dr.-Ing. Jürgen Beyerer

Eine Übersicht aller bisher in dieser Schriftenreihe
erschienenen Bände finden Sie am Ende des Buchs.

Ein Rahmenwerk für die Architektur von Frühwarnsystemen

von
Jürgen Moßgraber

Dissertation, Karlsruher Institut für Technologie (KIT)
Fakultät für Informatik, 2016

Impressum



Karlsruher Institut für Technologie (KIT)
KIT Scientific Publishing
Straße am Forum 2
D-76131 Karlsruhe

KIT Scientific Publishing is a registered trademark of Karlsruhe
Institute of Technology. Reprint using the book cover is not allowed.

www.ksp.kit.edu



*This document – excluding the cover, pictures and graphs – is licensed
under the Creative Commons Attribution-Share Alike 3.0 DE License
(CC BY-SA 3.0 DE): <http://creativecommons.org/licenses/by-sa/3.0/de/>*



*The cover page is licensed under the Creative Commons
Attribution-No Derivatives 3.0 DE License (CC BY-ND 3.0 DE):
<http://creativecommons.org/licenses/by-nd/3.0/de/>*

Print on Demand 2017 – Gedruckt auf FSC-zertifiziertem Papier

ISSN 1863-6489

ISBN 978-3-7315-0638-6

DOI 10.5445/KSP/1000066509

Ein Rahmenwerk für die Architektur von Frühwarnsystemen

zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

von der Fakultät für Informatik
des Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

Jürgen Moßgraber

aus Germersheim

Tag der mündlichen Prüfung:	21.12.2016
Erster Gutachter:	Prof. Dr.-Ing. Jürgen Beyerer
Zweiter Gutachter:	Prof. Dr. Sebastian Abeck

Danksagung

Die vorliegende Arbeit entstand während meiner Tätigkeit als Leiter der Forschungsgruppe Architekturen und Informationssysteme am Fraunhofer-Institut für Optronik, Systemtechnik und Bildverarbeitung IOSB.

Ich danke allen Personen, die mich zu dieser Arbeit ermutigt und mir während der Erstellung den Rücken gestärkt haben. Ganz besonderer Dank gilt meinem Betreuer Herrn Prof. Dr. Jürgen Beyerer für die Übernahme des Hauptreferates und seine wertvollen Anregungen zum Aufbau der Arbeit. Ebenso meinem Korreferenten Herrn Prof. Dr. Sebastian Abeck für die konstruktiven Diskussionen, die zu zahlreichen Hinweisen und Anregungen für die vorliegende Arbeit führten.

An dieser Stelle möchte ich mich auch bei allen Kolleginnen und Kollegen des Fraunhofer IOSB bedanken, insbesondere diejenigen der Abteilung Informationsmanagement und Leittechnik, mit denen ich in den letzten Jahren immer wieder an spannenden Themen forschen durfte. Besonderer Dank gilt meinem Abteilungsleiter Dr. Thomas Usländer, der mir den letzten Anstoß zu dieser Arbeit gegeben hat.

Großer Dank gebührt Frau Dr. Yvonne Fischer für das detaillierte Korrekturlesen (trotz sommerlicher Hitzewelle) und die daraus entstandenen zahlreichen kritischen Fragen und Kommentare.

Mein besonderer Dank gilt meiner Frau Eva und meinen Töchtern Annabelle und Sophie für die aufgebrachte Geduld und ihr Verständnis während der Erstellung dieser Arbeit.

Nicht zuletzt danke ich meinen Eltern für die lebenslange Förderung und Unterstützung, ohne die ich diese Arbeit nie geschrieben hätte.

Hördt, im September 2016

Jürgen Moßgraber

Inhaltsverzeichnis

Danksagung.....	i
1 Einleitung.....	1
1.1 Motivation	1
1.2 Problemstellungen.....	3
1.3 Zielsetzung und eigene Beiträge.....	6
1.4 Gliederung der Arbeit	10
2 Grundlagen	13
2.1 Katastrophenmanagement und Frühwarnung	13
2.1.1 Die Phasen des Katastrophenmanagements	18
2.1.2 Resilienz.....	23
2.1.3 Frühwarnung	25
2.1.4 Die Phasen der Frühwarnung	26
2.1.5 Die vier Bausteine effektiver Frühwarnung.....	28
2.1.6 Hauptbeteiligte und Interessengruppen	32
2.1.7 Einordnung der Frühwarnung	33
2.2 Systeme	34
2.2.1 Begriffsdefinitionen	34
2.2.2 Informationssysteme.....	41
2.2.3 Verteilte Informationssysteme.....	43
2.2.4 Entscheidungsunterstützungssysteme	47
2.2.5 Geographische Informationssysteme.....	50
2.2.6 Frühwarnsysteme.....	57
2.3 Systemarchitektur.....	58
2.3.1 Architektur.....	58
2.3.2 Definition des Systemarchitekturbegriffs.....	58
2.3.3 Architekturmuster	60
2.3.4 Bewertung von Architekturen	61
2.4 Beschreibung von Systemarchitektur	63

2.4.1	Modellierungssprachen.....	64
2.4.2	Rahmenwerke für die Systemarchitektur.....	88
2.5	Vorgehensmodelle.....	91
3	Stand von Forschung und Technik	93
3.1	Generische Basisarchitekturen.....	93
3.1.1	Dienstorientierte Architektur.....	93
3.1.2	Microservices	95
3.1.3	Ereignisgesteuerte Architektur.....	97
3.1.4	Event-Driven SOA	99
3.1.5	Lambda Architektur.....	100
3.2	Architekturen mit Schwerpunkt Sensorik.....	101
3.2.1	Internet of Things	101
3.2.2	Sensor Web Enablement	106
3.2.3	SANY	109
3.3	Architekturen zur Entscheidungsunterstützung.....	112
3.3.1	Architektur für Entscheidungsunterstützungssysteme	112
3.3.2	Architektur für Expertensysteme	114
3.4	Architekturen für Frühwarnsysteme	115
3.4.1	Generische Architekturen	115
3.4.2	Erdbeben	124
3.4.3	Tsunami	126
3.4.4	Flut.....	130
3.5	Zusammenfassung.....	133
4	Anforderungen an Frühwarnsysteme der nächsten Generation	135
4.1	Anforderungen an Frühwarnsysteme.....	135
4.1.1	Überwachung von Indikatoren.....	136
4.1.2	Prognose eines wahrscheinlichen Ereignisses.....	138
4.1.3	Kommunikation und Verbreitung von Warnungen	139
4.1.4	Reaktionsfähigkeit	140
4.2	Anforderungen zur Entscheidungsunterstützung.....	141
4.3	Anforderungen verteilter Systeme	142
4.3.1	Skalierbare Kommunikationsinfrastruktur für ein SoS	142

4.3.2	Resiliente Kommunikationsinfrastruktur für SoS	143
4.3.3	Effiziente Verwaltung großer Datenmengen.....	143
4.3.4	Skalierbare Speicherung großer verteilter Datenmengen	144
4.3.5	Sicherheit in verteilten Systemen.....	144
4.4	Anforderungen aus der Resilienz.....	144
4.5	Aggregierte Anforderungen.....	145
5	Ein Architekturrahmenwerk für Frühwarnsysteme	147
5.1	Unternehmenssicht	147
5.1.1	Hauptakteure	147
5.1.2	Rollen im Katastrophenfall	151
5.1.3	Analyse der Hauptakteure.....	152
5.1.4	Anwendungsfälle.....	153
5.1.5	Durchzuführende Schritte	154
5.2	System von Systemen Sicht	156
5.2.1	Prüfung, ob ein SoS vorliegt	156
5.2.2	Die Kommunikationsschicht für ein SoS.....	157
5.2.3	Durchzuführende Schritte	159
5.3	Systemsicht.....	159
5.3.1	Message-oriented Middleware	160
5.3.2	Datenquellen.....	161
5.3.3	Zwischenspeicher	161
5.3.4	Speicher.....	161
5.3.5	Wissensbasis.....	162
5.3.6	Verarbeitungsdienste	162
5.3.7	Simulationen.....	167
5.3.8	Verbreitung von Warnmeldungen.....	168
5.3.9	Workflow-Management-System	170
5.3.10	Semantischer Katalog.....	170
5.3.11	Geographisches Informationssystem	171
5.3.12	Benutzerschnittstelle.....	173
5.3.13	Durchzuführende Schritte	174
5.4	Informationssicht.....	174

5.4.1	Modellierung der Sensorik	175
5.4.2	Modellierung von Messwerten	175
5.4.3	Inhalt der MOM Nachrichten	176
5.4.4	Ontologie für die Wissensbasis	177
5.4.5	Durchzuführende Schritte	178
5.5	Technologiesicht	178
5.5.1	Message-oriented Middleware	178
5.5.2	Datenquellen	180
5.5.3	Zwischenspeicher	181
5.5.4	Datenarchiv	182
5.5.5	Wissensbasis.....	183
5.5.6	Verarbeitungsdienste und Simulationen.....	183
5.5.7	Verbreitung von Warnmeldungen.....	184
5.5.8	Workflow-Management-System	185
5.5.9	Semantischer Katalog	186
5.5.10	Geographisches Informationssystem	186
5.5.11	Benutzerschnittstelle.....	186
5.5.12	Durchzuführende Schritte	187
5.6	Konstruktionssicht	187
5.6.1	Interaktionsmatrix und Integrationstests.....	187
5.6.2	Verteilung der Komponenten	191
5.6.3	Sicherheit in verteilten Systemen.....	191
5.6.4	Durchzuführende Schritte	192
5.7	Gesammelte Arbeitsschritte	192
6	Einsatz semantischer Technologien in Frühwarnsystemen	195
6.1	Datenintegration	197
6.1.1	Datenveröffentlichung und -transformation.....	198
6.1.2	Semantische Integration von Informationssystemen	199
6.2	Verwaltung von Sensorik und Diensten.....	201
6.2.1	Schnittstellen zum semantischen Katalog	202
6.2.2	Implementierung des semantischen Katalogs.....	204
6.3	Verarbeitungsdienste	209

6.4	Wissensbasis	211
6.4.1	Analyse verfügbarer Ontologien	211
6.4.2	Modellierung der Domäne	215
6.4.3	Integration in SOA und EDA.....	216
6.5	Verbreitung von Warnmeldungen	221
6.6	Benutzerschnittstelle	225
7	Automatisierung von Arbeitsabläufen	227
7.1	Nutzung von BPMN.....	227
7.2	Das Workflow-Management-System.....	228
7.2.1	Architektur.....	228
7.2.2	Implementierung.....	230
7.3	Aufruf von Komponenten	231
8	Anwendungsbeispiele	233
8.1	Tsunami-Frühwarnung.....	233
8.1.1	Festlegung der Anwendungsfälle mit Hilfe von SOPs	234
8.1.2	Prüfung, ob ein SoS vorliegt	237
8.1.3	Implementierung einer Kommunikationsschicht	237
8.1.4	Integration aller notwendigen Datenquellen	237
8.1.5	Verarbeitungsdienste	238
8.1.6	Beschreibung der spezifischen Sensortypen	238
8.1.7	Softwarelösungen für jede Komponente	239
8.2	Sicheres Trinkwasser	241
8.2.1	Festlegung der Anwendungsfälle	243
8.2.2	Integration aller notwendigen Datenquellen	248
8.2.3	Verarbeitungsdienste	248
8.2.4	Prüfung, ob ein BigData Problem vorliegt	249
8.2.5	Anbindung aller Simulatoren.....	249
8.2.6	Benötigtes Kartenmaterial	251
8.2.7	Spezifikation der Benutzeroberfläche	252
8.2.8	Softwarelösungen für jede Komponente	254

9 Bewertung und Schlussbetrachtungen	257
9.1 Bewertung RA-FWS ^{NG}	257
9.1.1 Funktionale Anforderungen	258
9.1.2 Nichtfunktionale Anforderungen	259
9.1.3 Abgleich mit dem Stand von Forschung und Technik	260
9.2 Zusammenfassung	262
9.3 Ausblick	263
Abkürzungsverzeichnis.....	265
Abbildungen	269
Tabellen	275
Literaturverzeichnis (alphabetisch)	277
Eigene Veröffentlichungen (chronologisch).....	299

1 Einleitung

1.1 Motivation

Am 26. Dezember 2004 löste ein gewaltiges Erdbeben im Indischen Ozean eine Kaskade von Flutwellen aus. Im Osten des Epizentrums wurden Indonesien, Thailand und Malaysia ohne Vorwarnung von den Wassermassen getroffen – im Westen erreichten die Flutwellen Indien und Sri Lanka und überspülten die Malediven. Alleine in Sri Lanka starben 34.000 Menschen, da kein Tsunami-Frühwarnsystem vorhanden war [Zeit, 2014]. In der Bundesrepublik Deutschland litt im Jahr 2010 besonders das Land Sachsen unter wiederholten Hochwasserereignissen. Mit seinen verheerenden Überschwemmungen an Elbe und Donau hält das Jahr 2002 in Deutschland einen negativen Rekord für Flutschäden. Die Versicherer überwiesen ca. 1,8 Milliarden Euro an die Betroffenen [GDV, 2011].

Die Vereinten Nationen fordern deshalb schon lange den verstärkten weltweiten Aufbau von Frühwarnsystemen [UNISDR, 2005]. Diese Forderung wurde im Jahr 2015 in ihrer Resolution *The Sendai Framework for Disaster Risk Reduction 2015-2030* noch ausgeweitet auf die Berücksichtigung von verketteten Katastrophen [UNISDR, 2015].

Frühwarnsysteme dienen zur möglichst frühzeitigen Information über eine sich anbahnende oder auftretende Gefahr, um Personen und Organisationen die Möglichkeit zu geben, entsprechend darauf reagieren zu können [UNISDR, 2009]. Sie kommen deshalb in den unterschiedlichsten Bereichen vor, in denen hierzu eine Notwendigkeit besteht, wie beispielsweise bei durch die Natur oder den Menschen verursachte Katastrophen. Anwendungsfelder gibt es aber auch in der Industrie [Mayr, 2009], wenn zum Beispiel über einen drohenden Produktionsausfall wegen

fehlender oder fehlerhafter Teile informiert werden soll oder an der Börse bei drohenden Kursverlusten [Pettway, 1980].

In dieser Arbeit werden Anwendungsfelder behandelt, bei denen Menschenleben gefährdet sind. Dies trifft auf folgende Ereignisse zu:

- Geologische Ereignisse wie Erdbeben, Erdbeben, Tsunamis und Vulkanausbrüche [Wenzel, 2014]
- Unwetter wie Tornados, Hurrikans, Starkregen und Schneestürme
- Industrieunfälle, bei denen giftige Gase austreten oder Gewässer verunreinigt werden können
- Nukleare Katastrophen
- Epidemien und Pandemien wie Malaria, Cholera oder Ebola

Weitere Katastrophen wie Krieg und Terrorismus fallen auch unter diese Kategorisierung, werden in dieser Arbeit aber nicht betrachtet.

Katastrophen können sowohl einzeln, als auch in Kombination auftreten. Der schwere Tsunami in Japan im Jahr 2011 verursachte neben der großräumigen Überschwemmung die komplette Havarie der küstennahen Industrieanlagen, inklusive dem Atomkraftwerk Fukushima [French, 2013]. Bei diesem Kraftwerk kam es zu einer partiellen Kernschmelze mit einer unkontrollierten Freisetzung von radioaktiven Substanzen. Falls für diese unterschiedlichen Katastrophen einzelne Frühwarnsysteme vorhanden sind, müssen diese über geeignete Schnittstellen miteinander kommunizieren, da die einzelnen Auswirkungen nicht unabhängig voneinander sind.

Durch den Einsatz eines Frühwarnsystems können mögliche Auswirkungen abgewendet bzw. die Folgen der Katastrophe abgemildert werden. Um entscheiden zu können, ob eine echte Gefährdung vorliegt, müssen alle relevanten Informationen den jeweiligen Entscheidern zur Verfügung stehen. Ein entsprechendes Unterstützungssystem muss hierzu Daten

aggregieren, aufbereiten und in übersichtlicher Form präsentieren. Zum Verständnis der aktuellen Gefährdungslage und der Effektivität von Maßnahmen eignen sich zusätzlich Simulationen (Ausbreitungsrechnung eines Gefahrstoffs, Ankunft der Tsunamiwelle, etc.). Liegt eine reale Gefährdung vor, müssen entsprechende Maßnahmen zur Abwehr oder Minderung der Folgen in die Wege geleitet werden. Ein Frühwarnsystem muss also folgende Phasen abdecken: Messung und Sammlung der Daten, Auswertung und Überwachung (Grenzwerte), Bewertung und Erkennung von Gefahren, Verbreitung der Warnung, Vorschläge für Reaktionen zur Behebung/Vorsorge und schließlich die Meldung einer Entwarnung.

Bei Frühwarnsystemen ist der geographisch verteilte und kollaborative Charakter zu berücksichtigen. Da Katastrophen nicht an Ländergrenzen haltmachen, müssen die Messstationen über mehrere Länder verteilt sein. Die aggregierte Information muss an die länderspezifischen Entscheidungsinstanzen verteilt werden und auch die Warnung der Bevölkerung landesspezifisch erfolgen. Hierbei kommt zusätzlich noch die Problematik der Mehrsprachigkeit ins Spiel.

1.2 Problemstellungen

Im Folgenden werden die Problemstellungen (PS) beschrieben, die bei der Systemarchitektur von Frühwarnsystemen auftreten. Sie bilden die Motivation für die darauffolgende Zielsetzung und somit die Grundlage für die Beiträge der vorliegenden Arbeit.

PS1: Spezielle Architekturprobleme bei Frühwarnsystemen

Zunächst müssen die besonderen Herausforderungen bei der Architektur von Frühwarnsystemen verstanden werden. Moßgraber et al. diskutierten in [Moßgraber, 2013] die *sieben größten Herausforderungen bei der Definition einer Architektur von Frühwarnsystemen*:

1. Erstellung einer skalierbaren Kommunikationsinfrastruktur für ein System von Systemen (SoS, engl. System-of-Systems)
2. Erstellung einer resilienten Kommunikationsinfrastruktur für SoS
3. effiziente Verwaltung großer Datenmengen mit semantischen Annotationen
4. skalierbare und schnelle Speicherung von großen und verteilten Datenmengen
5. Handhabung heterogener Daten aus unterschiedlichen Domänen
6. Verwaltung und Auffinden von Sensorik in räumlich verteilten SoS
7. Koordination der Aufgaben in räumlich verteilten SoS

Bei dieser Veröffentlichung lag der Fokus auf der grundlegenden Infrastruktur eines solchen Systems. Hinzu kommen jedoch noch weitere Fragestellungen wie zum Beispiel:

- Wie werden Simulatoren integriert?
- Wie werden Arbeitsabläufe modelliert?
- Wie erfolgt die Information der Öffentlichkeit?

Spahn et al. [Wenzel, 2014] legen dar, dass ein Frühwarnsystem einem ständigen Lern- und Verbesserungsprozess unterliegt, der idealerweise in Zyklen aus Beurteilung, Planung und Konzeption, Implementierung und Auswertung durchgeführt wird. Für die Architektur ergibt sich daraus die Anforderung nach Flexibilität auf allen Ebenen, die von heutigen Systemen nur unzureichend erfüllt wird.

PS2: Generische Architektur für Frühwarnsysteme

Bestehende Lösungen für Frühwarnsysteme sind auf eine bestimmte Domäne zugeschnitten (Tsunami [Wächter, 2012], Erdbeben [Wenzel, 2014], Feuer [Marklund, 2016], etc.) und versuchen nicht eine Arbeitsumgebung vorzuschlagen, die die Entwicklung und den Betrieb von

Frühwarnsystemen im Allgemeinen erleichtert. Weiterhin entsteht der Bedarf nach mehr Allgemeingültigkeit und Flexibilität aus der Forderung nach der Berücksichtigung von verketteten Katastrophen [UNISDR, 2015].

Soll ein Frühwarnsystem konzipiert werden, stellt dies komplexe Herausforderungen an die Systemarchitekten [UNISDR, 2006]. Ein modernes Frühwarnsystem muss verschiedene (teilweise bereits existierende) Komponenten und Subsysteme wie verschiedene Sensornetzwerke, Informationsquellen, Dienstleistungen und Simulationssysteme integrieren. Hierdurch entsteht die Problematik eines SoS, für die besondere Maßnahmen getroffen werden muss [Baldwin, 2008]. Bei der Architektur eines Frühwarnsystems wird man mit der gesamten Bandbreite von Problemen der aktuellen Informatikforschung konfrontiert wie zum Beispiel der Analyse großer Datenmengen (BigData) [Fertier, 2016], dem Internet der Dinge (IoT, engl. Internet of Things) [Negahban, 2016] oder der Informationsgewinnung aus sozialen Netzwerken [Temnikova, 2015], die teilweise noch ungelöst sind [Moßgraber, 2013].

PS3: Management von Sensorik

An ein Frühwarnsystem sind zahlreiche Sensoren oder sogar vollständige Sensornetzwerke angebunden, um die relevanten Parameter einer entstehenden Gefahr zu überwachen. Diese eingesetzten Sensoren sind nicht statisch. Neue Sensoren kommen auf den Markt, die bessere Messwerte liefern, neue Parameter überwachen können oder auch eine intensivere Vorverarbeitung von Sensordaten ermöglichen, bzw. eine höhere Autonomie besitzen. Ein modernes und generisches Frühwarnsystem muss also eine große Menge an Sensoren verwalten können und die Fähigkeit besitzen, alte Sensoren zu entfernen und neue hinzuzufügen.

PS4: Abbildung von Datenquellen und Datenspeicherung

Die Daten, die von zahlreichen Sensoren geliefert werden, befinden sich in unterschiedlichen Datenformaten und besitzen auch nicht immer die gleiche

Semantik. Ein Frühwarnsystem muss deshalb die Fähigkeit besitzen, eine Datenintegration sowohl auf syntaktischer als auch auf semantischer Ebene durchführen zu können. Außerdem müssen die Daten für verschiedene Anwendungszwecke vorgehalten werden: detaillierte Daten für tiefgehende, meist langlaufende Analysen, Echtzeitdaten für Bewertungsalgorithmen und Analyseergebnisse zur Entscheidungsunterstützung.

PS5: Unterstützung dynamischer Standardvorgehensweisen und Abläufe

Sowohl für die Phase der Bewertung und Erkennung von Gefahren als auch in der Reaktionsphase eines Frühwarnsystems sind meist standardisierte Vorgehensweisen festgelegt. Diese werden mit dem englischen Begriff Standard Operating Procedure (SOP) oder auch Standard Operation Procedure bezeichnet. Ein generisches Frühwarnsystem muss mit der Problematik umgehen können, dass sich SOPs je nach Land oder Nutzergruppe stark unterscheiden können, d.h. SOPs müssen leicht ergänzt und modifiziert werden können [Wenzel, 2014].

Da Katastrophen und Krisensituationen glücklicherweise eher selten auftreten, ergibt sich die Problematik, dass die verantwortlichen Personen (Betreiber kritischer Infrastruktur, Behörden, Polizei, Rettungsdienste, etc.) wenig Praxiserfahrung vorweisen können und ihnen die notwendigen auszuführenden Aktionen nicht präsent sind. Dieser Problematik lässt sich auch durch Training nur begrenzt entgegenwirken. Dabei können Frühwarnsysteme unterstützen, indem sie vorab festgelegte Abläufe starten bzw. den Nutzer in den notwendigen Schritten begleiten und führen.

1.3 Zielsetzung und eigene Beiträge

Die aufgezeigten Problemstellungen, die im Kontext der Systemarchitektur von Frühwarnsystemen genannt wurden, stellen die Grundlage und Motivation für die vorliegende Arbeit dar. Ziel ist es daher, ein Rahmenwerk für die Architektur von Frühwarnsystemen zu entwickeln. Basierend auf der

Zielsetzung liefert die Arbeit in folgenden Bereichen einen wesentlichen Beitrag, die über den Stand der Forschung Technik hinausgehen.

B1: Identifikation der wesentlichen Architekturprobleme bei Frühwarnsystemen

Die Herausforderungen bei der Architektur eines Frühwarnsystems hängen stark vom gewünschten Einsatzzweck und der Anwendungsdomäne ab. Deshalb stehen das Verständnis der Domäne und die speziellen Problematiken an erster Stelle. Eine ausführliche Ausarbeitung für Frühwarnsysteme fehlte bisher in der Literatur. Dieser Beitrag adressiert die Problemstellung PS1.

B2: Erarbeitung eines Rahmenwerks für die Architektur von Frühwarnsystemen

Um Systemarchitekten bei der komplexen Aufgabe zu unterstützen, eine Systemarchitektur für Frühwarnsysteme zu entwerfen, wurde ein Rahmenwerk für die Architektur von Frühwarnsystemen der nächsten Generation (RA-FWS^{NG}) entwickelt. Dieses Rahmenwerk setzt auf einem bereits erprobten generischen Rahmenwerk für die Erstellung von Informationssystemen auf. Hierzu wurde das *Reference Model of Open Distributed Processing (RM-ODP)* ausgewählt und an die Bedürfnisse von Frühwarnsystemen angepasst. RM-ODP ist ein Metamodell zur Beschreibung von Informationssystemen, welches in der Norm der International Organization for Standardization (ISO) 10746 definiert ist [ISO, 2009]. Das Modell arbeitet mit verschiedenen Sichten auf ein System, sodass die folgenden Aspekte abgedeckt werden.

Die in B1 herausgearbeiteten Architekturprobleme wurden in die Unternehmenssicht von RM-ODP eingeordnet. Die Unternehmenssicht beschreibt das Geschäftsmodell aus Sicht der Endnutzer, die Informationssicht behandelt die Struktur, Bedeutung und Verarbeitung von Informationen, die Systemsicht geht auf Schnittstellen der verteilten

Bausteine (Komponenten) ein, die Konstruktionssicht kümmert sich um die verteilten Kollaborationen und Bereitstellung und schließlich definiert die Technologiesicht die konkrete Umsetzung der Architektur. RM-ODP sieht außerdem die Erweiterung durch eigene Sichten vor. Im erarbeiteten Rahmenwerk wurde die neue Sicht System von Systemen eingeführt, um diese spezielle Problematik bei Frühwarnsystemen zu adressieren.

In dieser Arbeit werden konkrete Lösungen für jede Sicht vorgestellt. Für die Systemsicht wurde eine Rahmenarchitektur erarbeitet, die alle Komponenten eines Frühwarnsystems und ihre Interaktionen enthält, um die aufgestellten Anforderungen zu erfüllen. Diese Architektur kann als Blaupause zukünftiger Frühwarnsysteme dienen und sieht auch die Integration existierender Frühwarnsysteme vor. Für die vorgesehenen Freiheitsgrade der Rahmenarchitektur werden für jede Sicht Handlungsanweisungen gegeben, wie diese ausgestaltet werden können. Dieser Beitrag fokussiert die Problemstellung PS2.

B3: Nutzung von semantischen Methoden und Technologien innerhalb des Rahmenwerks

Semantik spielt in den unterschiedlichsten Komponenten eines Frühwarnsystems eine Rolle. Semantische Technologien eignen sich zur Lösung verschiedener Architekturprobleme wie zum Beispiel für die semantische Modellierung der Domäne, die Verwaltung der Sensorik, die Zusammenführung heterogener Daten, die Speicherung des Wissens, die Navigation durch das Netz an Informationen und die Präsentation von Informationen.

Vorliegende Arbeit liefert zu mehreren dieser Punkte Beiträge. Für die Verwaltung und das Auffinden von Sensorik wird ein semantischer Katalog (engl. Semantic Registry) untersucht. Für das Zusammenführen heterogener Daten werden semantische Annotationen getestet. Die Modellierung einer geeigneten Ontologie stellt hierbei einen wesentlichen Beitrag dar.

Werden Ontologien eingesetzt, müssen diese auch gespeichert und verwaltet werden können. Als besondere Randbedingung stellt sich die Frage, wie diese in eine Ereignis- oder Serviceorientierte Architektur integriert werden können. Hierzu werden Lösungsmöglichkeiten und Optimierungen erarbeitet und aufgezeigt. Dieser Beitrag fokussiert die Problemstellung PS2 im Allgemeinen und die Problemstellungen PS3 und PS4 im Speziellen.

B4: Nutzung automatisierter Arbeitsabläufe innerhalb des Rahmenwerks

Sowohl in der Phase der Bewertung und Erkennung von Gefahren als auch in der Reaktionsphase eines Frühwarnsystems spielt die Automation von Arbeitsabläufen eine wichtige Rolle. Hier wird als Beitrag die Nutzung von Methoden zur Geschäftsprozessmodellierung aus dem Unternehmensumfeld auf Frühwarnsysteme übertragen: Arbeitsabläufe werden mittels einer Sprache zur Geschäftsprozessmodellierung, der *Business Process Model and Notation (BPMN)*, modelliert und durch ein Workflow-Management-System interpretiert.

Einzelne Schritte eines Arbeitsablaufs können asynchron auf getrennten, möglicherweise auch geografisch verteilten Systemen ablaufen. Möglichkeiten zur Lösung dieser Problematik werden aufgezeigt. Weiterhin wird die Integration von Simulatoren in Arbeitsabläufe zur Verbesserung des Situationsverständnisses diskutiert. Beispielsweise kann bei einem Frühwarnsystem zur Trinkwasserverunreinigung automatisch eine Ausbreitungsberechnung gestartet werden, wenn ein Sensor an einer bestimmten Stelle im Netzwerk Alarm schlägt. Dieser Beitrag fokussiert daher die Problemstellung PS2 im Allgemeinen und die Problemstellung PS5 im Speziellen.

Praxisrelevanz des Rahmenwerks

Die Nutzbarkeit des Rahmenwerks wurde in zwei Domänen erprobt und nachgewiesen: der Frühwarnung vor Tsunamis und der Frühwarnung vor verunreinigtem Trinkwasser.

1.4 Gliederung der Arbeit

Die inhaltliche Gliederung der Arbeit ist in Abbildung 1 dargestellt. Die Forschungsbereiche, die in das Rahmenwerk einfließen, sind mit unterschiedlichen Farben dargestellt. Die Kreise enthalten die jeweilige Kapitelnummer, in der das Thema besprochen wird.

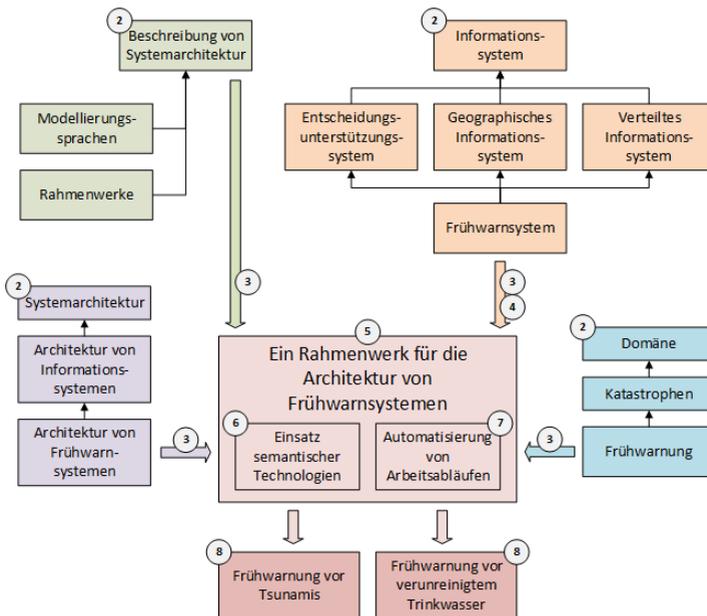


Abbildung 1: Forschungsgebiete, die in die Arbeit einfließen (unterschiedliche Farben), mit Kapitelübersicht (Zahlen in Kreisen)

In Kapitel 2 werden die notwendigen Grundlagen zur Domäne Frühwarnung, zu Informationssystemen im Allgemeinen und Frühwarnsystemen im Speziellen sowie zu Systemarchitekturen und deren Dokumentation gelegt. Danach werden existierende Rahmenwerke für die Architektur von Informationssystemen präsentiert.

Im anschließenden Kapitel 3 wird der Stand von Forschung und Technik erörtert. Dieser beinhaltet Forschungsarbeiten zu den Themen, die für das Verständnis der Architektur von Frühwarnsystemen notwendig sind. Zunächst wird die Systemarchitektur von Informationssystemen allgemein thematisiert und es werden spezielle Architekturarten wie SoS, Ereignis- und Serviceorientierte Architektur vorgestellt. Danach werden existierende Architekturen für Frühwarnsysteme bzw. Architekturen für Teilaspekte eines Frühwarnsystems analysiert und bewertet.

Kapitel 4 stellt die Anforderungen an die Architektur von Frühwarnsystemen auf. Das folgende Kapitel 5 beschreibt das entwickelte Rahmenwerk für die Architektur von Frühwarnsystemen. Das Kapitel gliedert sich entlang den verschiedenen Sichten von RM-ODP, wobei die neu eingeführte Sicht *System von Systemen* noch hinzukommt. Das Unterkapitel Systemsicht enthält die Beschreibung der eigentlichen Systemarchitektur, die alle Komponenten eines Frühwarnsystems enthält und ihre Interaktionen beschreibt.

In Kapitel 6 wird der Aspekt der Nutzung semantischer Modellierung und Technologien in Frühwarnsystemen vertieft. Lösungsmöglichkeiten für die generelle Modellierung von Semantik werden aufgezeigt. Weiterhin werden eine Ontologie für die Beschreibung und Verwaltung von Sensorik modelliert, ein Konzept für die Zusammenführung heterogener Daten mittels Annotationen erstellt und die effiziente und performante Speicherung von Ontologien erarbeitet.

Kapitel 7 vertieft die Nutzung von automatisierten Arbeitsabläufen innerhalb des Rahmenwerks. Gezeigt wird zunächst die Nutzung von BPMN für Frühwarnsysteme. Danach werden Implementierungsmöglichkeiten für

spezielle Arbeitsschritte und ein Konzept die Integration von Simulatoren erarbeitet. In Kapitel 8 wird das in den Kapiteln 5 bis 7 vorgestellte Rahmenwerk RA-FWS^{NG} auf zwei Domänen angewandt: der Frühwarnung vor Tsunamis und der Frühwarnung vor verunreinigtem Trinkwasser. Die Arbeit schließt in Kapitel 9 mit einer Bewertung, einer Zusammenfassung und einem Ausblick.

2 Grundlagen

2.1 Katastrophenmanagement und Frühwarnung

Naturgefahren wie Stürme, Dürren, Vulkanausbrüche oder Erdbeben müssen nicht zur Katastrophe führen. Eine Katastrophe tritt nur auf, wenn die Bewohner der betroffenen Region und ihre verantwortlichen Institutionen auf die Auswirkungen nicht entsprechend reagieren können.

Sintflutartige Regenwetter in der Mitte eines Ozeans verursachen keine Katastrophe, aber gehen die gleichen schweren Niederschläge auf eine Barackenstadt nieder, die an einem unbewachsenen Hang eines Hügels liegt, kann dies zu Erdbeben und einem großen Verlust von Menschenleben führen [ISDR, 2016]. Die Bevölkerung eines Tsunami-gefährdeten Küstenstreifens, der ein Frühwarnsystem fehlt, kann im Schlaf von einer Tsunamiwelle überrascht werden.

Der hierfür geprägte englische Begriff *Vulnerability* lässt eine Naturgefahr erst zur Katastrophe werden. Eine passende deutsche Übersetzung ist nicht vorhanden, mögliche Begriffe sind *Verwundbarkeit*, *Schwächen* oder *Schwachstellen* oder man greift auf *Vulnerabilität* zurück. Blaikie definiert Verwundbarkeit in folgender Weise [Blaikie, 2003].

„The characteristics of a person or group and their situation that influence their capacity to anticipate, cope with, resist and recover from the impact of a natural hazard (an extreme natural event or process).“

Das IFRC (International Federation of Red Cross and Red Crescent Societies) formuliert die Definition etwas prägnanter und erweitert gleichzeitig die Ursachen auf durch den Menschen verursachte Gefahren [IFRC, 1999].

„The characteristics of a person or group in terms of their capacity to anticipate, cope with, resist and recover from the impact of a natural or man-made hazard.“

Da in dieser Arbeit ebenfalls die durch den Menschen verursachten Katastrophen betrachtet werden, soll in Anlehnung an das IFRC folgende deutsche Definition gelten.

*Definition 1: **Verwundbarkeit** (Vulnerabilität) beschreibt die Leistungsfähigkeit einer Person oder einer Gruppe, die Auswirkungen einer durch die Natur oder den Menschen verursachten Gefahr vorausszusehen, zu bewältigen, ihnen zu widerstehen und sich schließlich davon zu erholen.*

Die Leistungsfähigkeit oder Kapazität (engl. Capacity) stellt also den Gegenpol zur Verwundbarkeit dar.

Abbildung 2 zeigt den Zusammenhang zwischen Verwundbarkeit, Gefahr und Katastrophe auf. Eine Katastrophe wird umso wahrscheinlicher je höher das Risiko ist. Ein Risiko besteht bei vorhandener Verwundbarkeit durch eine Gefahr. Eine Katastrophe wird ausgelöst durch ein auftretendes Ereignis wie beispielsweise ein Erdbeben oder eine Flut.

Weiterhin nennt Blaikie Beispiele für die Ursachen der Verwundbarkeit und gliedert diese in drei Stufen [Blaikie, 2003].

1. Grundlegende Ursachen: Armut, beschränkter Zugriff auf Ressourcen, Ideologien, ökonomische Systeme
2. Dynamischer Druck: Mangel an lokaler Verwaltung, Ausbildung, Training, Bevölkerungsdruck, Urbanisierung
3. Gefährliche Rahmenbedingungen: Unsichere Bauorte, schwache Gebäude und Infrastrukturen, schwache Wirtschaft

Diese führen in ihrer Kombination zum Anstieg der Verwundbarkeit. Aus Abbildung 2 lässt sich die Definition für den Begriff der *Katastrophe* ableiten.

*Definition 2: Eine **Katastrophe** entsteht aus einem vorhandenen hohen Risiko durch das Auftreten eines auslösenden Ereignisses. Ein Risiko besteht bei vorhandener Verwundbarkeit für eine oder mehrere Gefahren.*

Zur Abwendung von Katastrophen muss die Leistungsfähigkeit der Bevölkerung und Institutionen gestärkt werden.

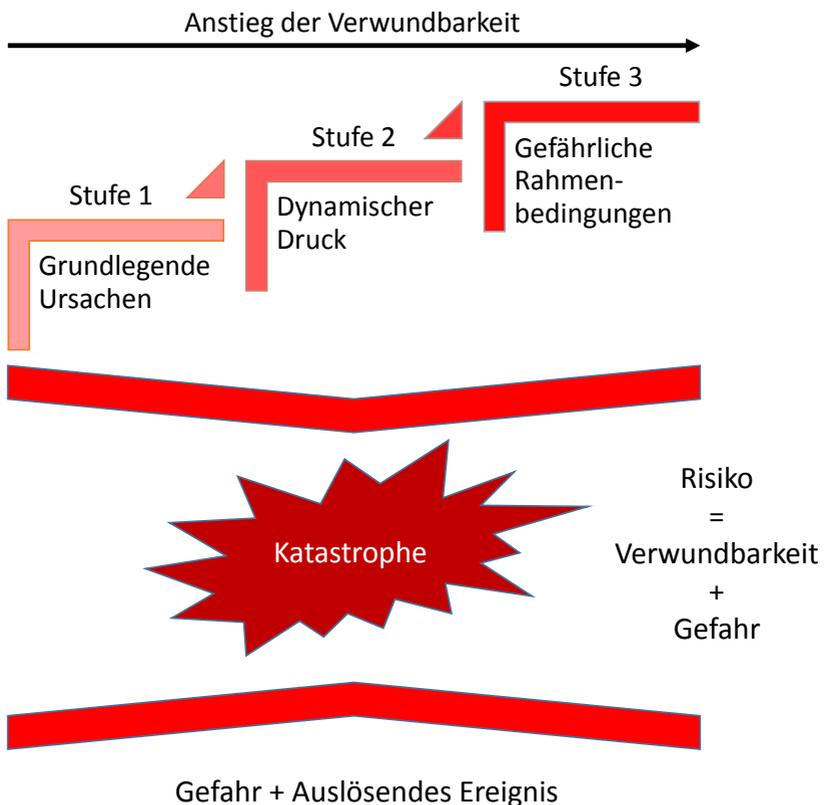


Abbildung 2: Verwundbarkeit, Gefahr und Katastrophe nach [Blaikie, 2003]

Hierzu ist zunächst eine Analyse der Verwundbarkeit und der aktuellen Leistungsfähigkeit notwendig. In diesem Zusammenhang spricht man von einer Bewertung der Verwundbarkeit und Leistungsfähigkeit (VCA, engl. Vulnerability and capacity assessment) [IFRC, 1999].

*Definition 3: Die **Bewertung der Verwundbarkeit und Leistungsfähigkeit** ist ein Prozess, der zur Identifizierung der Stärken und Schwächen von Haushalten, Gemeinschaften, Institutionen und Nationen dient [IFRC, 1999].*

Als Erweiterung der Definition auf die Umsetzung in die Praxis wird vom Konzept der Verminderung des Katastrophenrisikos (engl. Disaster risk reduction) gesprochen.

*Definition 4: Die **Verminderung des Katastrophenrisikos** ist das Konzept und die Praxis der Verminderung von Katastrophenrisiken durch systematische Analyse und Verwaltung der ursächlichen Faktoren von Katastrophen, unter anderem durch reduzierte Exposition gegenüber Gefahren, eine geringere Anfälligkeit von Personen und Eigentum, kluge Verwaltung von Land und Umwelt sowie die Verbesserung der Vorbereitung auf unerwünschte Ereignisse [IFRC, 2012].*

Führt man eine Analyse der Verwundbarkeit der Bevölkerung durch, müssen im Wesentlichen folgende beiden Fragen beantwortet werden:

1. Für welche Bedrohung oder Gefahr besteht eine Verwundbarkeit?
2. Was macht sie verwundbar für diese Bedrohung oder Gefahr?

Sind diese identifiziert und wurde die Problematik verstanden, können Gegenmaßnahmen in den folgenden Bereichen getroffen werden:

1. *Reduzierung der Auswirkungen* der Gefahr soweit dies möglich ist (durch Schadensbegrenzung, Vorhersage, Warnung und Vorsorge-maßnahmen).

2. *Aufbau von Kapazitäten* zum Widerstehen von und zum Umgang mit Gefahren.
3. *Beseitigung der grundsätzlichen Ursachen* der Verwundbarkeit wie Armut, schlechte Verwaltung oder unzureichender Zugriff auf Ressourcen.

Die konkrete Organisation von Aufgaben und Abläufen obiger Punkte fallen in den Bereich des *Katastrophenmanagements*.

Definition 5: Als **Katastrophenmanagement** bezeichnet man Maßnahmen einer Organisation in Reaktion auf unerwartete Ereignisse, die Menschen oder Ressourcen beeinträchtigen und die Fortsetzung des Betriebes gefährden. Darin enthalten sind Vorschläge zur Erholung nach der Katastrophe. Diese Gesundungskonzepte sollen die Risiken durch Katastrophen minimieren und wirksame Gegenmaßnahmen ermöglichen. Katastrophenmanagement bezieht sich im Regelfall auf Naturkatastrophen wie Brände, Überschwemmungen und Erdbeben [Onpulson, 2016].

Im Englischen entspricht Katastrophenmanagement dem Wort *Disaster Management*. Aktuell wird jedoch von internationalen Rettungshelferorganisationen der Begriff *Emergency Management* bevorzugt, der sich mit Notfallmanagement übersetzen lässt. Dieser Begriff passt besser zur obigen Definition 2. Eine Krise oder ein Notfall ist eine bedrohliche Situation, die umgehende Maßnahmen erfordert. Wirksame Notfallmaßnahmen können die Eskalation eines Ereignisses vermeiden, sodass daraus keine Katastrophe entsteht.

Notfallmanagement beinhaltet Pläne und institutionelle Regelungen zur Unterstützung der Bemühungen von Regierungs- und nicht Regierungsorganisationen sowie freiwilligen und privaten Rettungsdiensten, um in umfassender Weise auf das gesamte Spektrum der dringendsten Bedürfnisse reagieren zu können [UNISDR, 2009].

Im Folgenden wird jedoch der im Deutschen weiterhin gebräuchlichere Begriff Katastrophenmanagement beibehalten.

2.1.1 Die Phasen des Katastrophenmanagements

Das Katastrophenmanagement wird häufig in vier Phasen eingeteilt (s. Abbildung 3 und Abbildung 4):



Abbildung 3: Phasen des Katastrophenmanagements [Austrian, 2011]

1. Vorsorge (Planung) / Vermeidung / Abschwächung (engl. *Mitigation*)
2. Vorsorge (Training) / Vorbereitung (engl. *Preparedness*)
3. Bewältigung (engl. *Response*)
4. Wiederherstellung / Nachbereitung (engl. *Recovery*)

Während sich im Englischen die vier Begriffe *Mitigation*, *Preparedness*, *Response* und *Recovery* fest eingebürgert haben, findet man im Deutschen häufig unterschiedliche Bezeichnungen. Phase 1 und 2 werden auch zu *Prevention* zusammengefasst [Ammann, 2016].

Das Konzept der *Phasen* wurde bereits seit den 1930er Jahren verwendet, um Katastrophen zu beschreiben, zu untersuchen und zu helfen, die Praxis des Katastrophenmanagements zu organisieren. Neal zitiert Beispiele mehrerer Forscher die fünf bis acht Phasen anführen, lange bevor die vier Phasen zum Standard wurden [Neal, 1997]. Die Grundlage für die verbreitete Verwendung der vier englischen Phasen ist das Ergebnis der Arbeit der *National Governors' Association* (NGA) in den USA der späten 1970er Jahren [NGA, 1979].



Abbildung 4: Der Kreislauf des Katastrophenmanagements [NEHRP, 2016]

Unabhängig von der Begrifflichkeit sind die Inhalte der Phasen jedoch meist ähnlich:

1. Phase: Risikoabschätzungen, Katastrophenplanung, Krankheitsprävention, Gesundheitsförderung
2. Phase: Schulungen und Training, interdisziplinäre Kooperationen, bauliche Maßnahmen, Information und Training der Bevölkerung, sonstige Vorbereitungen

3. Phase: Rettung von Menschenleben, Medizinische und psychosoziale Notfallversorgung für die Bevölkerung, Schutz von Gebäuden und der Umwelt, Aufbau der Notfallversorgung
4. Phase: Wiederherstellung des *Normalzustands*, wie zum Beispiel der Wiederaufbau von Gebäuden und der Infrastruktur, Nachbereitung

2.1.1.1 Verzahnung und Parallelität der Phasen

In den letzten Jahren hat sich die Erkenntnis durchgesetzt, dass die Phasen nicht strikt voneinander getrennt sind [Baird, 2010]. Stattdessen werden die Phasen verzahnt dargestellt (s. Abbildung 4). Man spricht dann vom Kreislauf des Katastrophenmanagements (engl. Emergency Management Cycle oder Disaster Management Cycle) oder kurz vom *Krisenkreislauf*.

Neuere Veröffentlichungen stellen auch die strikte Abfolge der Phasen in Frage (s. Abbildung 5) und sehen sie eher als parallel stattfindende Vorgänge bzw. auszuführende Tätigkeiten an [FEMA, 2009].

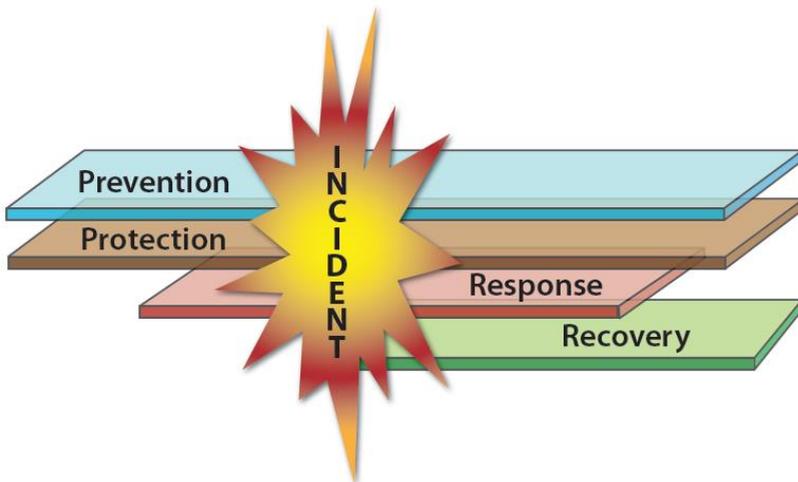


Abbildung 5: Parallelität der Phasen des Katastrophenmanagements [FEMA, 2009]

Kritische Aktivitäten decken also häufig mehr als eine Phase ab, und die Grenzen zwischen den Phasen sind nicht präzise. Baird vergleicht und diskutiert in *The Phases of Emergency Management* die unterschiedlichen Phasen, ihre Inhalte und ihre Interaktion, wie sie in den diversen Organisationen der USA im Einsatz sind sowie ihre Veränderung bzw. Entwicklung im Lauf der Jahre [Baird, 2010].

Dass auch in Deutschland keine Einigkeit über die Einteilung der Phasen herrscht, zeigt die Recherche nach dem Begriff *Katastrophenmanagement-* oder *Krisenmanagementzyklus*, welche zahlreiche Darstellungsvarianten mit unterschiedlichen Begrifflichkeiten liefert. Beispielhaft sei der Krisenkreislauf angeführt, der vom Zentrum für Satellitengestützte Kriseninformation, angesiedelt beim Deutschen Zentrum für Luft- und Raumfahrt (DLR), verwendet wird (s. Abbildung 6).

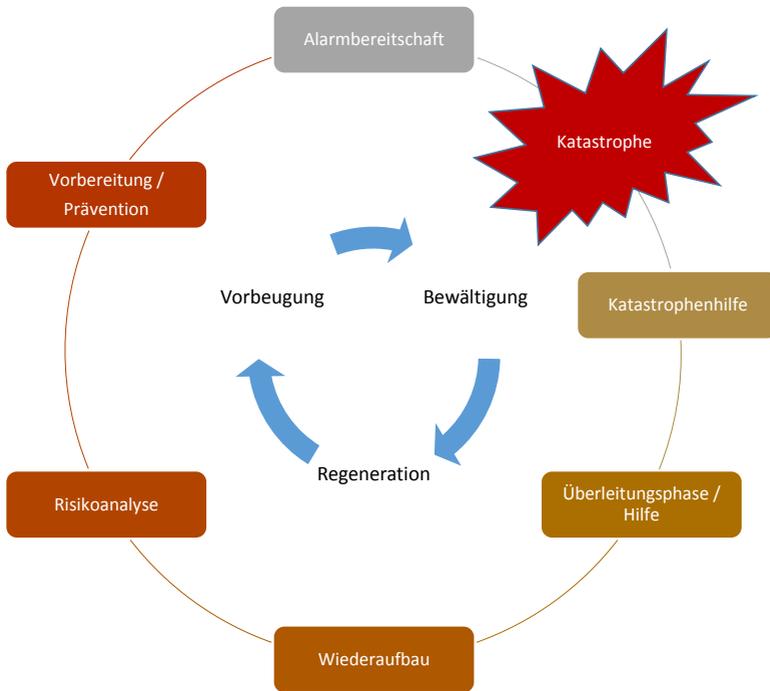


Abbildung 6: Krisenkreislauf nach [ZKI, 2013]

In dieser Darstellung wird die Tätigkeit *Risikoanalyse* als eigene Phase verstanden. Statt einer Verzahnung der Phasen wird außerdem nur eine *Überleitungsphase* zwischen der Bewältigungsphase (hier *Katastrophenhilfe* genannt) und dem Wiederaufbau eingefügt.

Im Folgenden wird aufgrund obiger Diskussion die Einteilung in die vier Phasen mit ihren englischen Begriffen *Mitigation*, *Preparedness*, *Response* und *Recovery* verwendet.

2.1.1.2 Standard Operating Procedure

Sowohl für die Phase der Bewertung und Erkennung von Gefahren als auch in der Reaktionsphase eines Frühwarnsystems sind meist standardisierte Vorgehensweisen festgelegt. Diese werden mit dem englischen Begriff Standard Operating Procedure (SOP) oder auch Standard Operation Procedure bezeichnet. Der Begriff kann auf Deutsch etwa mit Standardvorgehensweise, standardisiertes Vorgehen oder Standardarbeitsanweisung übersetzt werden, wird aber selten gebraucht. SOPs sind als unverzichtbare Voraussetzung des Qualitätsmanagements im ISO Standard 9001 spezifiziert [ISO, 2015]. Tabelle 1 zeigt eine Vorlage zur textuellen Beschreibung einer SOP.

Tabelle 1: Aufbau einer SOP

Ziel und Zweck	Was soll warum erreicht werden?
Anwendung	Arbeitsbereich
Beschreibung des Ablaufs	Wie und womit wird die Tätigkeit durchgeführt?
Zuständigkeit, Qualifikation	Wer ist zuständig, wer darf durchführen?
Dokumentation	Was, wo, wie, durch wen?

2.1.2 Resilienz

Der Begriff Resilienz (engl. Resilience) ist in den letzten Jahren sehr häufig zu lesen. Resilienz wird als gesellschaftliches und staatliches Konstrukt und als Schutzkonzept kritischer Infrastrukturen erforscht. Konkrete Forschungsschwerpunkte sind der resiliente Bürger, Energiesicherheit und resiliente Industrieunternehmen [Thoma, 2014]. Der Fokus der Risikoforschung (wie können Gefahren vermieden werden?) wechselt in Richtung Resilienz (wie können die Auswirkungen minimiert werden?).

Grob gesprochen wird das Katastrophenmanagement auf eine allgemeinere Ebene gehoben. Dementsprechend wurde der Kreislauf des Katastrophen-

managements (s. Abbildung 4) adaptiert. Abbildungen 7 und 8 zeigen Beispiele für die Adaption, die nun Resilienz-Zyklus genannt wird. Die Phasen werden ebenfalls mit den Begrifflichkeiten des Katastrophenmanagements bezeichnet.

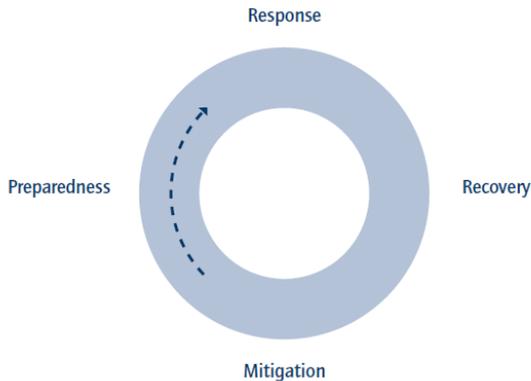


Abbildung 7: Der Social Resilience Cycle [Edwards, 2009]

Die in Abschnitt 2.1.1.1 diskutierte Problematik der Verzahnung und Parallelität der Phasen wird überraschenderweise bei Resilienz noch nicht thematisiert. Jedoch kritisieren Häring et al., dass während und nach jeder Phase neue Ereignisse eintreten können, die den Ablauf des Zyklus stören (*disruptive Ereignisse*) und auf die entsprechend reagiert werden muss [Häring, 2016]. Obwohl die Thematik nicht neu ist, lenkt sie die Forschung auf einen wichtigen Aspekt: heutigen Systemen mangelt es an der Flexibilität, um auf die Anforderungen der Resilienz eingehen zu können.

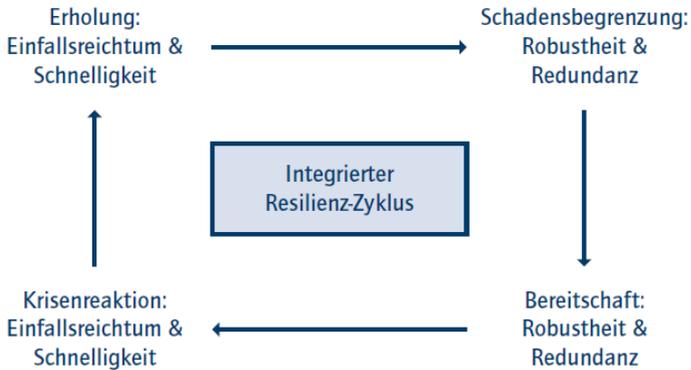


Abbildung 8: Resilienz-Zyklus nach [Trachslers, 2009]

2.1.3 Frühwarnung

Die International Federation of Red Cross and Red Crescent Societies (IFRC) und das United Nations Office for Disaster Risk Reduction (UN/ISDR) verwenden für den Begriff der Frühwarnung die folgende Definition.

*Definition 6: **Frühwarnung** ist die Menge der erforderlichen Kapazitäten (Leistungsfähigkeit), die benötigt wird, um rechtzeitige und sinnvolle Warnungen zu erzeugen und an Einzelpersonen, Gemeinschaften und Organisationen zu verbreiten, die von einer Gefahr bedroht werden, damit sie eine angemessene und ausreichende Vorlaufzeit haben, um sich darauf vorzubereiten und somit einen möglichen Schaden oder Verlust zu reduzieren oder sogar zu vermeiden [UNISDR, 2009], [IFRC, 2012].*

Diese Definition fordert indirekt eine Reihe von Faktoren, die notwendig sind, um wirksame Reaktionen auf die ausgeschickten Warnungen zu erreichen. Ein Frühwarnsystem umfasst notwendigerweise vier Schlüsselemente: die Kenntnis der Risiken, deren Überwachung, die Analyse und Prognose der Gefahren, die Kommunikation oder Verbreitung

von Meldungen und Warnungen sowie die lokalen Kapazitäten, um auf diese Warnungen entsprechend zu reagieren. Hierfür wird auch gerne der Ausdruck *Ende-zu-Ende-Warnsystem* (engl. End-to-end Warning System) verwendet, um zu betonen, dass ein Warnsystem alle Schritte von der Gefahrenerkennung bis hin zur Reaktion der Gemeinschaft umfasst.

In der englischen Originaldefinition wird statt des Begriffs Frühwarnung (engl. Early Warning) der Begriff Frühwarnsystem (EWS, engl. Early Warning System) verwendet. Damit ist jedoch nicht ein technisches Informationssystem bestehend aus Hard- und Software gemeint, sondern ein holistischer Ansatz, der alle Aspekte der Frühwarnung umfasst. Da es im Kern dieser Arbeit um Frühwarnsysteme im technischen Sinne geht, die diesen ganzheitlichen Prozess unterstützen, wird im Folgenden der Begriff Early Warning System von IFRC und UN/ISDR mit *Frühwarnung* übersetzt, um Verwirrung zu vermeiden. Die Bemühungen des UN/ISDR zielen auf die starke Integration und Mitwirkung der Bevölkerung im Frühwarnprozess ab, da dies nach ihrer Einschätzung ein unerlässlicher Faktor für den Erfolg der Frühwarnung ist. Dies spiegelt sich auch in den Forderungen und Fragestellungen wieder. Um diese Einschätzung weiter herauszustellen, sprechen UN/ISDR als auch IFRC nun von Community Early Warning Systems [IFRC, 2012]. Für den Begriff des Frühwarnsystems soll folgende Definition gelten:

*Definition 7: Ein **Frühwarnsystem** unterstützt den ganzheitlichen Prozess der Frühwarnung mit technischen Lösungen aus Hard- und Software.*

Daraus ergibt sich die Frage, welche Schritte oder Phasen dieser ganzheitliche Prozess umfasst.

2.1.4 Die Phasen der Frühwarnung

Das traditionelle Rahmenwerk der Frühwarnung besteht aus drei Phasen:

1. Überwachung von Indikatoren (Vorboten)
2. Prognose eines wahrscheinlichen Ereignisses
3. die Benachrichtigung mittels einer Warnung oder eines Alarms im Falle, dass ein Ereignis katastrophalen Ausmaßes stattfindet

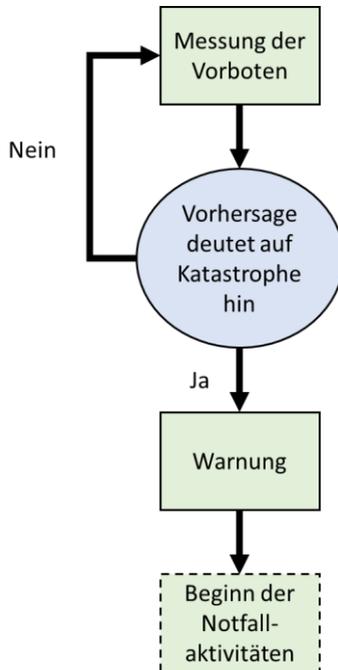


Abbildung 9: Das Vier-Phasen Modell der Frühwarnung nach [Basher, 2006]

Ein verbessertes Vier-Stufen-Rahmenwerk wird durch nationale und internationale Katastrophen- und Risikomanagementinstitutionen befürwortet (s. Abbildung 9). Diese zusätzliche vierte Phase umfasst den Beginn der Notfallaktivitäten, sobald eine Warnung ausgegeben wurde. Der Zweck dieses neuen Abschnitts ist der Erkenntnis geschuldet, dass die zu erfolgende Warnung ins Gesamtkonzept integriert werden muss, während

man sich in der Vergangenheit für diese Tätigkeit auf die Ersthelferorganisationen verlassen hat [Basher, 2006].

Ähnlich wie beim Katastrophenmanagement ist die strikte Abgrenzung der Phasen kritisch zu sehen. Diese Problematik wird in der Literatur nur wenig diskutiert [Bell, 2010]. Die Erkenntnis, dass die Phasen eng verzahnt sind bzw. sogar parallel verlaufen können, scheint aber dennoch vorhanden zu sein, da die internationale Strategie zur Katastrophenvorsorge der Vereinten Nationen nicht mehr von Phasen sondern von den vier Bausteinen der Frühwarnung spricht, die stark vernetzt sind [ISDR, 2016].

Diese vier Bausteine sind:

1. Verständnis der Risiken
2. Warndienst
3. Kommunikation und Verbreitung von Warnungen
4. Reaktionsfähigkeit

2.1.5 Die vier Bausteine effektiver Frühwarnung

Eine gute Frühwarnung hat starke Verknüpfungen zwischen den vier Bausteinen. Wird ein Baustein nicht richtig umgesetzt, kann das zum Versagen des gesamten Prozesses führen. Indikatoren für gute Frühwarnkonzepte sind die Folgenden [ISDR, 2016]:

- Die wichtigsten Akteure, die an den jeweiligen Bausteinen beteiligt sind, müssen sich regelmäßig treffen, um die Problematiken der anderen Komponenten zu verstehen und zu besprechen, was von ihnen zugeliefert werden muss.
- Risikoszenarien werden gemeinsam erstellt und evaluiert. Für alle Aufgaben im gesamten Prozess werden die Zuständigkeiten vereinbart und umgesetzt.
- Historische Ereignisse werden untersucht und daraus Verbesserungen für die Frühwarnung abgeleitet.

- Verfahren werden in Handbüchern dokumentiert und veröffentlicht.
- Die Bevölkerung wird in die Risikoanalyse einbezogen und Erkenntnisse und Informationen werden mit ihr geteilt.
- Notwendige Maßnahmen wie beispielsweise Evakuierungen werden geübt und getestet.

All diese Aktivitäten können nur funktionieren, wenn eine solide Basis aus politischer Unterstützung, Gesetzen und Verordnungen, klarer institutioneller Verantwortung und ausgebildeten Menschen vorhanden ist [ISDR, 2016].

2.1.5.1 Wissen um die Risiken

Dieser Baustein beschreibt das systematische Sammeln von Daten und die Durchführung von Risikobewertungen [ISDR, 2016]. Risiken ergeben sich sowohl aus Gefahren als auch aus vorhandenen Schwächen (Vulnerabilitäten). Was sind die Muster und Trends dieser Faktoren? Risikobewertung und ihre Abbildung auf die Praxis tragen dazu bei, die Anforderungen an die Frühwarnung zu priorisieren und die Vorbereitungen für die Reaktions- und Präventionsaktivitäten zu unterstützen [Basher, 2006]. Die Risikoabschätzung kann auf historischen Erfahrungen sowie menschlichen, sozialen, wirtschaftlichen und ökologischen Schwachstellen basieren.

Fragestellungen in diesem Baustein sind:

- Sind alle Gefahren und die Schwachstellen bekannt und verstanden?
- Was sind die Muster und Trends in diesen Faktoren?
- Sind Risikokarten und Daten überall verfügbar?

2.1.5.2 Überwachungs- und Warndienst

Dieser Baustein beschreibt die Durchführung der Entwicklung der Gefahrenüberwachung und der Implementierung von Frühwarndiensten [ISDR, 2016].

Für die Vorhersage von potenziell katastrophalen Ereignissen ist eine solide wissenschaftliche Grundlage erforderlich. Oft gibt es Warnhinweise auf eine drohende Gefahr lange bevor sie eintritt. Beispielsweise beobachten Meteorologen den Beginn von Hurrikannen und Stürmen und berechnen daraus ihre wahrscheinliche zukünftige Stärke und Route. Vulkanforscher interpretieren ein verdächtiges Grollen unter der Erde, Ökologen bewerten die Wahrscheinlichkeiten für den Auftritt von Heuschreckenplagen, Dürre-Experten schätzen die Chancen für Niederschläge, Ozeanographen beobachten die Tiefen des pazifischen Ozeans für Anzeichen des El Niño und verfolgen die Ausbreitung von durch Erdbeben verursachten Tsunamiwellen [ISDR, 2016]. Jeder dieser Forschungszweige hat hierzu zahlreiche *Indikatoren*. Eine ständige Überwachung der identifizierten Indikatoren ist notwendig, um genaue Warnungen rechtzeitig erzeugen zu können. Ansätze, die viele Gefahren adressieren und viele Beobachtungsstationen enthalten, werden als am effektivsten erachtet [Basher, 2006].

Fragestellungen in diesem Baustein sind:

- Werden die richtigen Parameter überwacht? Was sind die passenden Frühwarnindikatoren?
- Gibt es eine solide wissenschaftliche Grundlage zur Erstellung der Prognosen?
- Können genaue und rechtzeitige Warnungen generiert werden?

2.1.5.3 Kommunikation und Verbreitung von Warnungen

Dieser Baustein beschreibt die Kommunikation von Informationen über Risiken und frühe Warnungen zu Gefahren [ISDR, 2016].

Es ist wichtig, dass Warnhinweise die gefährdeten Personen erreichen. Damit die Menschen die Warnungen verstehen, müssen diese klar formuliert sein und nützliche Informationen enthalten, die die richtige Reaktion ermöglichen. Passende Kommunikationskanäle müssen auf regionaler, nationaler und auf Ebene der Bevölkerung im Voraus identifiziert und etabliert werden [Basher, 2006].

Fragestellungen in diesem Baustein sind:

- Erreichen die Warnungen alle gefährdeten Personen?
- Werden die Risiken und die Warnungen verstanden?
- Sind die in der Warnung enthaltenen Informationen und Empfehlungen klar und nutzbar?

2.1.5.4 Reaktionsfähigkeit

Dieser Baustein beschreibt den Aufbau von Reaktionsfähigkeiten auf nationaler Ebene und auf Ebene der Bevölkerung [ISDR, 2016].

Es ist wichtig, dass die Bevölkerung die vorhandenen Risiken versteht, die Meldungen des Warndienstes ernst nimmt und weiß, wie man im Ernstfall zu reagieren hat. Der Aufbau einer informierten Bevölkerung erfordert die Beteiligung aller Wissensträger sowohl der offiziellen Bildungseinrichtungen als auch von ehrenamtlichen und privaten Initiativen [Basher, 2006].

Fragestellungen in diesem Baustein sind:

- Sind Reaktions- und Katastrophenpläne auf dem neuesten Stand und getestet?
- Wird von den lokal vorhandenen Kapazitäten und dem Wissen Gebrauch gemacht?

- Sind die Menschen entsprechend vorbereitet und bereit, auf Warnungen zu reagieren?

2.1.6 Hauptbeteiligte und Interessengruppen

Viele Beteiligte sind wichtig, damit Katastrophenmanagement und auch die Frühwarnung funktioniert: Beamte, Städte und Gemeinden, Wirtschaft, Regierungs- und Nichtregierungsorganisationen, Wissenschaftler, Akademiker, Lehrer, Medien und natürlich die Bevölkerung. Die besten Frühwarnsysteme finden Wege, um all diese Gruppen zu vernetzen und ihre Zusammenarbeit zu erleichtern [ISDR, 2016].

Da etwa 80% aller Katastrophen das Wetter betreffen, ist in den meisten Ländern der nationale meteorologische und hydrologische Dienst die wichtigste nationale Institution für die Veröffentlichung von Frühwarnungen. Manchmal trägt diese Behörde auch die Verantwortung für die Warnung vor Vulkan- und Erdbebengefahren. In einigen Ländern ist für Überschwemmungen und hydrologische Vorhersagen eine andere Behörde zuständig, zum Beispiel ein Amt für Wasserbau. Um Verwirrung zu vermeiden, ist es deshalb wichtig, dass eine eindeutige Stelle im Falle einer ernststen Gefahr für die Herausgabe von Meldungen zuständig ist.

In vielen Ländern gibt es zusätzliche Abteilungen bei Behörden, die für den Katastrophenschutz oder das Notfallmanagement im Krisenfall zuständig sind. Diesen oder anderen Behörden kann auch die Verantwortung für die Beurteilung der sozialen und wirtschaftlichen Auswirkungen der potentiellen Gefahren zufallen. Manchmal können sie auch die Prävention, Vorsorgeaktivitäten und die Öffentlichkeitsarbeit übernehmen [ISDR, 2016].

2.1.6.1 Katastrophenschutz in Deutschland

In Deutschland wird die Umsetzung des Katastrophenmanagements als *Katastrophenschutz* bezeichnet. Katastrophenschutz ist grundsätzlich Ländersache. Die Rolle des Bundes im Bereich Bevölkerungsschutz

beschränkt sich derzeit bei überregionalen Schadenereignissen auf eine Koordinierungs- und Ressourcenmanagementfunktion. Eine Weisungsbefugnis des Bundes gegenüber den Ländern ist nicht vorhanden. Ob die Koordinationsfunktion, wie sie seit 2009 im Zivilschutzänderungsgesetz Eingang gefunden hat, bei katastrophalen Ereignissen mit mehreren betroffenen Bundesländern ausreichend ist, kann aktuell nicht abschließend beantwortet werden [Gerhold, 2011].

Zum Katastrophenschutz tragen öffentliche Einrichtungen wie die Bundesanstalt Technisches Hilfswerk (THW), das Bundesamt für Bevölkerungsschutz und Katastrophenhilfe (BBK), Feuerwehren, Ordnungs- und Sicherheitsbehörden sowie private und kommunale Organisationen wie das Deutsche Rote Kreuz (DRK) oder der Arbeiter-Samariter-Bund (ASB) bei.

Prof. Lange merkte beim Workshop *Naturereignisse & Naturkatastrophen* an, dass sich der Bevölkerungsschutz in Deutschland mit seinen vielseitigen Akteurskonstellationen in Bund, Ländern, Kommunen, Privatwirtschaft, Interessenverbänden und Wissenschaft durch eine enorme Heterogenität und Undurchsichtigkeit auszeichnet. Weiterhin bezweifelte er, ob alle Akteure sich ihrer Rolle bzw. Aufgabenwahrnehmung innerhalb der Gefahrenabwehr bewusst sind [Gerhold, 2011].

2.1.7 Einordnung der Frühwarnung

Bisher wurde noch nicht angesprochen, welche Phase(n) des Katastrophenmanagements von der Frühwarnung unterstützt bzw. abgedeckt werden.

Das erste Baustein Verständnis der Risiken lässt sich der Phase Mitigation zuordnen, die Bausteine Warndienst, Kommunikation und Verbreitung von Warnungen der Phase Preparedness und die Reaktionsfähigkeit fällt in die Phase Response [Wenzel, 2014]. D.h., einzig die Phase des Wiederaufbaus, Recovery, scheint nicht die Frühwarnung zu betreffen. Zumindest müssen

aber die Erkenntnisse der in ihr enthaltenen Nachbereitungsanalyse wieder in die Aktualisierung der Risikoanalyse einfließen.

2.2 Systeme

2.2.1 Begriffsdefinitionen

Für den Begriff des *Systems* gibt es zahlreiche Definitionen, unter anderem auch in Standards des DIN [DIN, 1988] und der ISO [ISO, 1995]. Für die Betrachtungen in dieser Arbeit soll die folgende Definition gelten, da sie sowohl die „Umgebung“ und den „Kontext“ berücksichtigt als auch die mögliche Rekursivität von Systemen (System von Systemen):

*Definition 8: Ein **System** ist eine Menge von Elementen, die in einer bestimmten Umgebung oder in einem bestimmten Kontext als Einheit aufgefasst werden. Die Elemente dieser Einheit interagieren, stehen miteinander in Beziehung und können ihrerseits wiederum als System aufgefasst werden [Richling, 2011].*

Die Begriffe *Element* und *System* können durch diese Definition also synonym verwendet werden. Statt *Element* wird auch häufig das Wort *Komponente* verwendet.

2.2.1.1 Systemeigenschaften

In der Planungsphase der Systementwicklung werden Anforderungen (engl. Requirements) an ein System definiert, dies ist die Anforderungsanalyse (engl. Requirements Engineering). Die Definition des Begriffs *Anforderung* ist nicht trivial. Veröffentlichungen zur Anforderungsanalyse geben keine klare Definition, sondern führen die Definition oft über die Klassifikation von Anforderungen ein. Eine Ausnahme stellt Rupp dar, der den Begriff *Anforderung* als „eine Aussage über eine zu erfüllende Eigenschaft oder zu

erbringende Leistung eines Produktes, Prozesses oder der am Prozess beteiligten Personen“ definiert [Rupp, 2007]. Booch dagegen definiert eine Anforderung schlicht als „eine Voraussetzung oder eine Fähigkeit, die ein System erfüllen muss“ [Booch, 1999]. Das V-Modell [Bröhl, 1995] verlangt in der Anforderungsanalyse die Festlegung messbarer Eigenschaften, Forderungen und Merkmale und ergänzt, dass Anforderungen auf Erfüllung prüfbar sein müssen. In Anlehnung an Booch und Rupp soll folgende schlichte Definition gelten.

*Definition 9: Eine **Anforderung** ist eine Aussage über eine zu erfüllende Eigenschaft des Systems.*

Nun muss natürlich definiert werden, was eine Systemeigenschaft ist.

*Definition 10: **Systemeigenschaften** sind Eigenschaften, die ein System oder ein Element leistet und besitzt, um ein Problem zu lösen oder ein Ziel zu erreichen (vgl. [IEEE, 1990] und [Sommerville, 2012]).*

In der Literatur werden meist Anforderungen diskutiert (zum Beispiel bei der Klassifikation von Anforderungen), statt die sich daraus ergebenden Eigenschaften des Systems. In Anlehnung an Richling [Richling, 2011] wird im Folgenden einheitlich der Begriff Eigenschaft verwendet.

2.2.1.2 Klassifikation von Eigenschaften

In der Literatur wird meist als oberste Grobklassifikation zwischen *funktionalen* und *nichtfunktionalen* Eigenschaften unterschieden [Pohl, 2008]. Funktionale Eigenschaften beschreiben die Dienste und Funktionalitäten, die ein System bereitstellt. Nichtfunktionale Eigenschaften sind in der Literatur nicht einheitlich definiert, werden jedoch vielfach als Randbedingungen und Qualitätseigenschaften verstanden [Eigner, 2012]. Richling definiert nichtfunktionale Eigenschaften indirekt über die Dienstgüte (QoS, engl. Quality of Service), die sich aus den Messgrößen der nichtfunktionalen Eigenschaften ergibt [Richling, 2011]. Dieser Zu-

sammenhang wird in Abbildung 10 dargestellt. Funktionale Eigenschaften sind spezifisch für ein System. Nichtfunktionale Anforderungen dagegen wiederholen sich regelmäßig bei unterschiedlichen Systemen, weswegen es sinnvoll ist, diese zu klassifizieren. Zur Klassifikation gibt es in der Literatur verschiedene Arbeiten, die meist den Schwerpunkt auf einen bestimmten Aspekt setzen. Avižienis beispielsweise betrachtet die Verlässlichkeit [Avižienis, 2001], [Avižienis, 2004], Pohl den Bereich der Sicherheit [Pohl, 2008] und Richling die Autonomie [Richling, 2011].

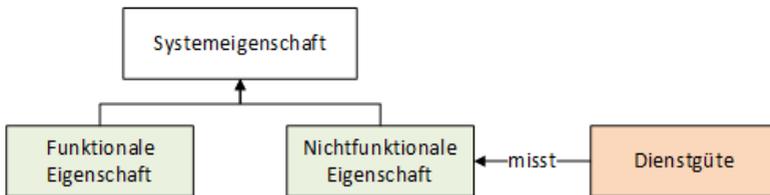


Abbildung 10: Klassifikation von Systemeigenschaften nach [Richling, 2011]

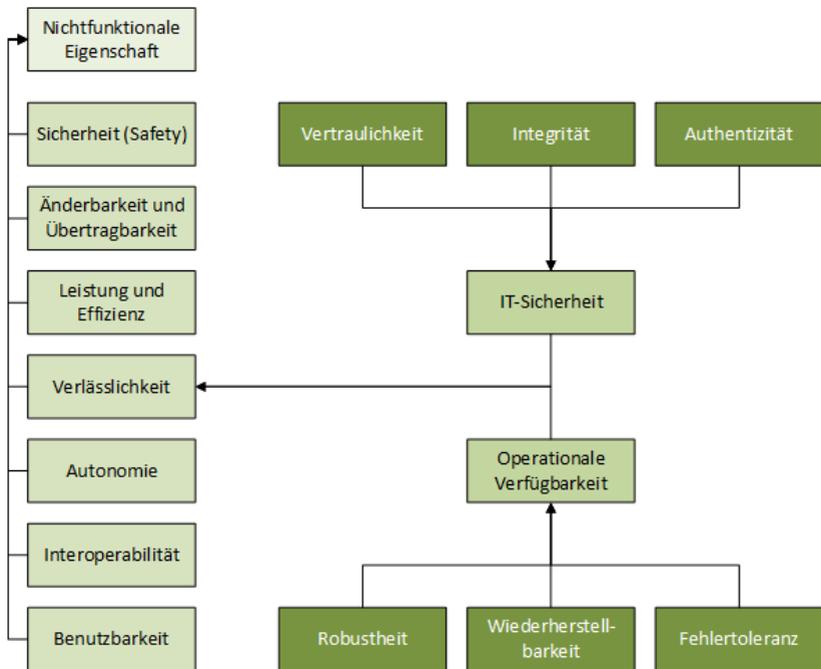


Abbildung 11: Klassifikation Nichtfunktionaler Systemeigenschaften nach [Avizienis, 2004], [Pohl, 2008], [ISO, 2011]

Die ISO wiederum klassifiziert in ihrem Standard *System und Software-Engineering – Qualitätskriterien und Bewertung von System und Softwareprodukten (SQuaRE) – Qualitätsmodell und Leitlinien* [ISO, 2011] System- und Softwareeigenschaften nach Qualitätseigenschaften.

Abbildung 11 führt die Ideen obiger Quellen in einer Hierarchie zusammen. Tabelle 2 enthält die zugehörigen Beschreibungen der Systemeigenschaften mit deutschen und englischen Bezeichnern.

Tabelle 2: Beschreibung der Systemeigenschaften

Deutsch	Englisch	Beschreibung
Sicherheit	Safety	Das System darf Menschen nicht gefährden.
Änderbarkeit und Übertragbarkeit	Maintainability	Aufwand, der zur Durchführung vorgegebener Änderungen notwendig ist und ggf. von der (Laufzeit-) Umgebung in eine andere übertragen werden zu können.
Leistung und Effizienz	Performance and Efficiency	Verhältnis zwischen dem Leistungsniveau der Software und dem Umfang der eingesetzten Betriebsmittel unter festgelegten Bedingungen.
Verlässlichkeit	Dependability	Die Eigenschaft, einen Dienst anzubieten, der vertrauenswürdig eingestuft wird.
Autonomie	Autonomy	Grad bis zu dem das System eigenständig Entscheidungen aufgrund vorliegender Erkenntnisse und zu erwartenden Ergebnissen treffen kann.
Interoperabilität	Interoperability	Fähigkeit, mit vorgegebenen Systemen zusammenzuwirken.
Benutzbarkeit	Usability	Aufwand, der zur Benutzung erforderlich ist und individuelle Beurteilung der Benutzung durch eine festgelegte oder vorausgesetzte Benutzergruppe.
IT¹-Sicherheit	Security	Fähigkeit, unberechtigten Zugriff sowohl versehentlich als auch vorsätzlich auf Programme und Daten zu verhindern.
Vertraulichkeit	Confidentiality and Privacy	Vermeidung der Herausgabe von unautorisierten Informationen.

¹ Informationstechnik

		Berücksichtigung datenschutzrechtlicher Aspekte.
Integrität	Integrity	Vermeidung fehlerhafter bzw. schadhafter Veränderungen des Systems.
Verfügbarkeit	Availability	Bereitschaft des Systems zur Erbringung seiner Dienste.
Wiederherstellbarkeit	Recoverability	Fähigkeit, bei einem Versagen das Leistungsniveau wiederherzustellen und die direkt betroffenen Daten wieder zu gewinnen. Zu berücksichtigen sind die dafür benötigte Zeit und der benötigte Aufwand.
Robustheit und Fehlertoleranz	Robustness and Fault Tolerance	Fähigkeit, ein spezifiziertes Leistungsniveau bei Fehlern oder Nicht-Einhaltung ihrer spezifizierten Schnittstellen zu bewahren.

2.2.1.3 Komposition

Das Zusammenfügen von Elementen zu einem neuen System führt zu dem Begriff der *Komposition*:

*Definition 11: **Komposition** bezeichnet das Zusammenfügen von Elementen (d.h. elementaren Elementen oder bereits existierenden Systemen) zu einem neuen System [Richling, 2011].*

Die Komposition von Elementen kann deren Eigenschaften im System verändern [Richling, 2003]:

- Invariante Eigenschaft: Die Eigenschaft bleibt unverändert und identifizierbar (also an eines der Elemente gebunden) erhalten
- Gebundene Eigenschaft: Die Eigenschaft wird an einen neuen Wert oder Wertebereich gebunden, bleibt aber einem der Elemente zuordenbar

- Verschwindende Eigenschaft: Die Eigenschaft kann im neuen System nicht mehr identifiziert werden
- Auftauchende Eigenschaft: Eine neue Eigenschaft entsteht (durch die Kombination von Einzeleigenschaften der Elemente)
- Übertragene Eigenschaft: Die Eigenschaft wird aus einer Komponenteneigenschaft zu einer Systemeigenschaft

2.2.1.4 Resilienz

Der Begriff Resilienz in Bezug auf Systeme ist nicht zu verwechseln mit dem holistischen Ansatz, der in Abschnitt 2.1.2 beschrieben wurde. Resiliente Systeme sollen widerstandsfähiger gegen Fehler sein. In der Literatur werden fünf Fehlertypen unterschieden [Tanenbaum, 2006]:

1. Absturzfehler: Ein System antwortet permanent nicht mehr.
2. Auslassungsfehler: Ein System beantwortet einzelne Anfragen nicht, lässt diese also aus.
3. Antwortzeitfehler: Die Antwortzeiten eines Systems liegen außerhalb eines festgelegten Zeitintervalls.
4. Antwortfehler: Das System liefert fehlerhafte Antworten.
5. Byzantinischer/zufälliger Fehler: Ein System gibt zu zufälligen Zeiten zufällige Antworten (verhält sich nichtdeterministisch)

Die Messgröße für die Resilienz ist die Verfügbarkeit (engl. Availability) eines Systems, die entsprechend der Verfügbarkeit aus der industriellen Produktion definiert ist als [Hanmer, 2007]:

$$\text{Verfügbarkeit} = \frac{\text{MTTF}}{\text{MTTF} + \text{MTTR}}$$

Wobei MTTF (engl. Mean Time To Failure) für die durchschnittliche Zeit vom Beginn des ordnungsgemäßen Betriebs eines Systems bis zum Auftreten eines Fehlers steht. MTTR (engl. Mean Time To Recovery) dagegen ist die durchschnittliche Zeit vom Auftreten eines Fehlers bis zur Wiederherstellung des ordnungsgemäßen Systembetriebs. Klassischerweise versucht man in

der Entwicklung zur Erhöhung der Verfügbarkeit die Anzahl der Fehler zu minimieren und somit ebenfalls den MTTF-Faktor zu maximieren. Bei Resilienz dagegen wählt man einen anderen Blickwinkel und betrachtet Fehler als unvermeidbar. Gerade bei großen verteilten Systemen sind Fehler der Normalfall und nicht die Ausnahme. Um die Verfügbarkeit aber trotzdem maximieren zu können, muss der Faktor MTTR minimiert werden.

*Definition 12: **Resilienz** ist die Fähigkeit eines Systems, mit unerwarteten (Fehler-)Situationen umzugehen, bestenfalls ohne dass der Anwender etwas davon mitbekommt, im schlechtesten Fall mit einer definierten Herabsetzung der Servicequalität (engl. Graceful Degradation of Service).*

2.2.2 Informationssysteme

Eine eindeutige Definition des Begriffs Informationssystem² ist in der Literatur überraschenderweise nicht zu finden. Existierende Definitionen sind außerdem stark vom Hintergrund des jeweiligen Autors geprägt. Eine Definition aus dem Bibliothekswesen lautet:

*Definition 13: Ein **Informationssystem** ist ein System, das der Speicherung und Wiedergewinnung von Information dient. Information liegt in informetrischen Einheiten vor. Ein besonderer Typ informetrischer Einheiten sind die Dokumente [Ewert, 1997].*

Abgesehen von dem ungebräuchlichen Begriff *informatrisch* besagt diese Definition, dass ein Informationssystem Informationen verwaltet und ihre Recherchierbarkeit unterstützt. Heinrich wählt eine knappe Definition, der er jedoch eine längere Erklärung über MAT-Systeme folgen lässt:

² Ein Diskurs über den Begriff *Information* sowie den Unterschied zwischen Daten und Information würde in dieser Arbeit zu weit führen.

Definition 14: *Informationssysteme* sind Mensch/Aufgabe/Technik-Systeme (MAT-Systeme) [Heinrich, 2011].

Knapp zusammengefasst wird hier der Mensch als Interagierender mit dem System beschrieben. Weiterhin dient das Informationssystem nicht nur zur Informationsabfrage, sondern unterstützt allgemein bei der Erfüllung seiner Aufgaben.

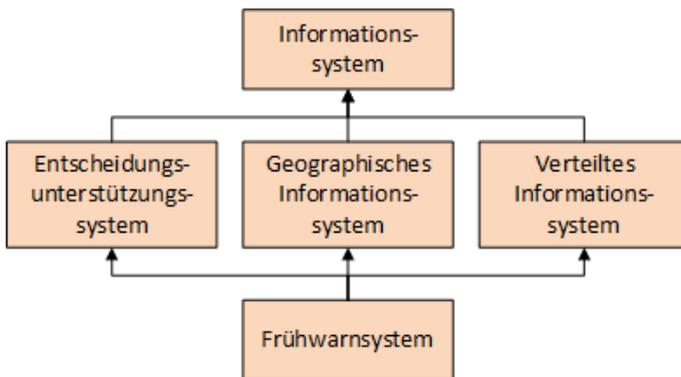


Abbildung 12: Oberkategorien von Frühwarnsystemen

Abbildung 12 zeigt die Unterkategorien von speziellen Informationssystemen bis zum Typ des Frühwarnsystems. Diese Kategorien werden in den folgenden Abschnitten beschrieben. Erwähnenswert ist zuvor noch die Kategorie von behördlichen oder betrieblichen *Umweltinformationssystemen*, da sie wichtige Datenquellen für Frühwarnsysteme darstellen können. Als Unterkategorie von Geographischen Informationssystemen werden die für Frühwarnsysteme relevanten Funktionen und Schnittstellen allerdings bereits im Abschnitt über geographische Informationssysteme (GIS) vorgestellt, deshalb werden sie nicht explizit in der Abbildung aufgeführt.

2.2.3 Verteilte Informationssysteme

Die Entwicklung verteilter Systeme begann ca. 1980 mit dem Aufkommen von lokalen Netzwerken (LAN, engl. Local Area Network). Die Bedeutung von verteilten Systemen stieg durch die möglich gewordene Kopplung mehrerer Rechner mittels LANs und der Tatsache, dass deren Preis-/Leistungsverhältnis sich im Laufe der Zeit im Vergleich zu Großrechnern stetig verbesserte, kontinuierlich an [Bengel, 2002].

In der Literatur findet sich eine Vielzahl unterschiedlicher Definitionen für verteilte Systeme. Gemeinsam haben alle Definitionen, dass sie versuchen, eine Abgrenzung zu dem im Gegensatz stehenden zentralistischen System zu finden. Dies gelingt oftmals aber nicht, da die Grenze zwischen verteilten und zentralistischen Systemen fließend ist. Exemplarisch hierfür ist die Definition von einem verteilten System als System von miteinander verbundenen und zusammenarbeitenden Prozessoren, da hier auch Rechner mit mehreren Prozessoren als verteilte Systeme gelten würden [Tanenbaum, 1985]. Diese sind aber eindeutig zentralistischen Systemen zuzuordnen. Es ergibt sich somit die Problematik, die Definition möglichst allgemein gültig zu halten, aber dennoch eine klare Abgrenzung zu zentralistischen Systemen zu wahren [Diekmann, 2003]. Folgende Definition stellt deshalb ein Kompromiss dar:

*Definition 15: Ein **verteiltes System** ist ein informationsverarbeitendes System, das eine Vielzahl von eigenständigen Rechnern enthält, die über ein Kommunikationsnetzwerk miteinander kooperieren, um ein angestrebtes Ziel zu erreichen [Bapat, 1994].*

2.2.3.1 Das CAP-Theorem

Das CAP-Theorem besagt, dass ein verteiltes System gleichzeitig nur zwei der drei Eigenschaften Konsistenz, Verfügbarkeit und Partitionstoleranz

gleichzeitig erfüllen kann. Gilbert und Lynch führten dazu einen axiomatischen Beweis [Gilbert, 2002].

- Konsistenz (engl. Consistency): In einem verteilten System, das mit replizierten Daten arbeitet, muss sichergestellt sein, dass nach Abschluss einer Transaktion auch alle Replikate des geänderten Datensatzes aktualisiert werden.
- Verfügbarkeit (engl. Availability): Hier ist nicht die unter 2.2.1.4 definierte Verfügbarkeit gemeint, sondern die Verfügbarkeit im Sinne akzeptabler Antwortzeiten des Systems. Alle Anfragen an das System werden stets beantwortet.
- Partitionstoleranz (engl. Partition Tolerance): Die Ausfalltoleranz der Rechner-/Servernetze. Das System arbeitet auch bei Verlust von Nachrichten, einzelner Netzknoten oder Partition des Netzes weiter.

2.2.3.2 System-of-Systems

Es existiert keine allgemein akzeptierte Definition für den Begriff System von Systemen (SoS, engl. System-of-Systems), aber die meisten Definitionen decken die folgenden Eigenschaften ab, die Baldwin in *Systems Engineering Guide for Systems of Systems* aufgestellt hat [Baldwin, 2008]:

„A System of Systems (SoS) is defined as a set or arrangement of systems that results when independent and useful systems are integrated into a larger system that delivers unique capabilities. Both individual systems and SoS conform to the accepted definition of a system in that each consists of parts, relationships, and a whole that is greater than the sum of the parts; however, although a SoS is a system, not all systems are SoS.“

Daraus ergibt sich folgende frei übersetzte Definition:

Definition 16: Ein System von Systemen (SoS) ist eine Anordnung von Systemen, die sich ergibt, wenn voneinander unabhängige Systeme in ein größeres System integriert werden. Dieses bietet neue Funktionalitäten, die in den Einzelsystemen nicht vorhanden sind. Sowohl die Einzelsysteme wie auch das SoS entsprechen der Definition eines Systems (s. Definition 8), obwohl jedoch ein SoS ein System ist, sind nicht alle Systeme ein SoS.

Außer der Eigenschaft, dass nicht alle Systeme ein SoS sind gibt die Definition keine echten Kriterien für die Entscheidung, ob wir mit einem SoS konfrontiert sind, impliziert aber die Absicht, neue Fähigkeiten zu entwickeln durch Synergien mehrerer Legacy-Systeme. Maier stellt in *Architecting principles for systems-of-systems* fünf Hauptmerkmale für die Identifikation eines echten SoS [Maier, 1998]:

1. Operative Unabhängigkeit der Elemente: Wenn das SoS in seine Bestandteile zerlegt wird, müssen die Komponentensysteme in der Lage sein, selbständig zu arbeiten. Das SoS besteht aus Systemen, die unabhängig voneinander und selbstständig einen Zweck erfüllen können.
2. Unabhängigkeit der Elemente in der Verwaltung: Die Komponentensysteme müssen nicht nur unabhängig voneinander arbeiten können, sie arbeiten unabhängig, d.h. die Komponentensysteme werden zwar integriert, behalten aber eine kontinuierliche betriebliche Existenz unabhängig von dem übergreifenden SoS.
3. Evolutionäre Entwicklung: Ein SoS ist nicht bereits von Beginn an in vollem Umfang spezifiziert. Seine Entwicklung und Existenz ist evolutionär. Ihm werden Funktionen und Zwecke hinzugefügt, entfernt und mit steigender Erfahrung angepasst.

4. Emergentes³ Verhalten: Das System führt Funktionen und Zwecke aus, die sich in keinem der Komponentensysteme befinden. Diese Verhaltensweisen sind emergente Eigenschaften des gesamten SoS. Die Hauptzwecke des SoS zeichnen sich durch diese Eigenschaften aus.
5. Geographische Verteilung: Die geographische Verteilung der Teilsysteme ist groß. *Groß* ist ein relativer Begriff bei der heutigen zunehmenden weltweiten Vernetzung, deshalb wird als Anforderung ergänzt, dass die Komponenten nur Informationen und nicht größere Mengen an Masse oder Energie austauschen können.

SoS Engineering befasst sich mit der Planung, Analyse, Organisation und der Integration der Fähigkeiten einer Mischung aus bestehenden, sich in der Entwicklung oder noch in Planung befindlicher unabhängiger Systeme.

Schnittstellen und Kommunikation

Ein wichtiger Teil der Architektur des SoS ist die Spezifikation, wie die Systeme zusammenarbeiten. Für SoS, die auf Informationsaustausch angewiesen sind, konzentriert sich die Definition der Schnittstellen auf die Art des Informationsaustauschs der Teilsysteme [Baldwin, 2008]. Bei diesen Systemen besteht die Notwendigkeit, gemeinsame Kommunikationsmechanismen zu definieren. Ebenso wichtig ist die Definition der Syntax und Semantik der gemeinsamen oder gemeinsam genutzten Daten. Die Schnittstellen der Systeme müssen weiterhin die Koordination des Systemverhaltens sowie die Aktionen, die für das kollektive Verhalten der Teilsysteme im SoS notwendig sind, umfassen. Typischerweise besitzt die SoS-Architektur einen strukturierten Ansatz, der beschreibt, wie die Systeme miteinander in Beziehung stehen und das Hinzufügen / Ersetzen von Systemen oder Funktionen für die Evolution der SoS ermöglicht.

³ Vom englischen Wort *emergent* für *auftauchend* bzw. *erscheinend*.

Maier sieht die Spezifikation der Schnittstellen nicht nur als eines der grundlegenden Prinzipien beim Entwurf von SoS an sondern als das einzige [Maier, 1998]:

„Wenn die Komponenten eines SoS unabhängig sind, sowohl operativ als auch in der Verwaltung, besteht die Architektur des SoS einzig aus seinen Schnittstellen. Es gibt nichts anderes zu entwerfen. Das Internet ist die Schnittstelle, in diesem Fall das Internet Protocol (IP).“

Somit ist kein SoS möglich ohne Kommunikation. Daher ist eine lose Kopplung der Teilsysteme erforderlich und Änderungen dürfen nicht zu einem funktionsunfähigen SoS führen.

Entwicklung und Evolution

Die oben angeführte Verwaltung der Teilsysteme durch verschiedene Parteien impliziert, dass die Planung- und Entwicklungszyklen parallel stattfinden. Dies wurde von Garlock näher untersucht und ist in Abbildung 13 dargestellt [Garlock, 2001]. In dieser Abbildung sind drei Teilsysteme dargestellt, die in ein SoS integriert werden. Für alle Teilsysteme sind die ersten beiden Entwicklungsphasen dargestellt. Die Eigenschaften der Teilsysteme können nun erst zu den gewünschten Funktionalitäten X, Y und Z des SoS kombiniert werden, wenn die Entwicklungsphasen der Teilsysteme abgeschlossen sind.

2.2.4 Entscheidungsunterstützungssysteme

Ein Entscheidungsunterstützungssystem (DSS, engl. Decision Support System) ist ein Softwaresystem, das für menschliche Entscheidungsträger zur Bewältigung ihrer operativen und strategischen Aufgaben relevante Informationen ermittelt, aufbereitet, übersichtlich zusammenstellt und bei der Auswertung hilft. Dazu gehören Funktionen zur Sortierung und Filterung von Daten, ihre flexible Darstellung sowie Auswertungsmöglichkeiten wie

Summen, Durchschnittsberechnungen oder Vergleiche. Weitergehende Funktionen erlauben die Ausführung von Modellrechnungen (Szenarien und Prognosen) und die Verknüpfung der Daten mit Optimierungsalgorithmen [Keen, 1980].

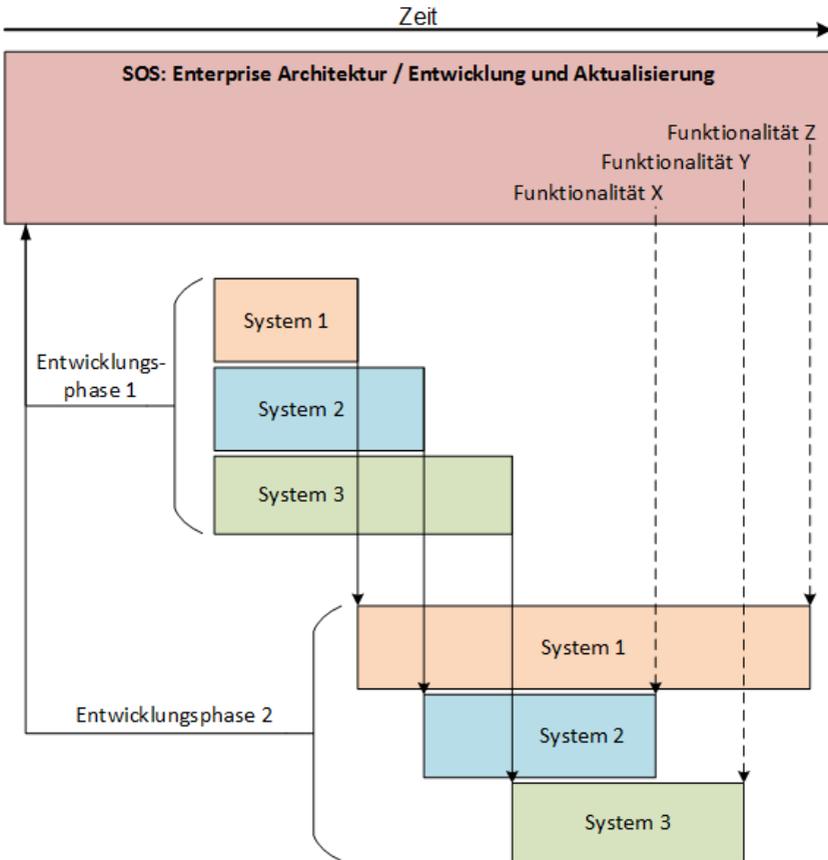


Abbildung 13: SoS-Architektur- und Entwicklungszyklen nach [Garlock, 2001]

Sprague definiert ein DSS durch seine Eigenschaften [Sprague, 1980]:

- DSS dient oft zur Unterstützung der oberen Führungsebene bei der Lösung von weniger gut strukturierten, unterspezifizierten Problemen.
- DSS versucht, die Modelle und Analysetechniken mit traditionellen Datenzugriffstechniken und Abruffunktionen zu kombinieren.
- DSS konzentriert sich speziell auf Funktionen, die vor allem Menschen, die nicht technikaffin sind, die interaktive Verwendung ermöglichen.
- DSS besitzt die Flexibilität sich an Veränderungen aus der Umwelt und an die spezielle Arbeitsweise des Benutzers anzupassen.

Häufig findet man in der Literatur auch die beiden Begriffe Beratungssystem (engl. Advisory System) und Expertensystem (engl. Expert System) die als spezialisierte Unterkategorien von DSS zu sehen sind. Sowohl Beratungssysteme als auch Expertensysteme sind Pakete zur Problemlösung, die einen menschlichen Experten in einem spezialisierten Fachbereich nachahmen. Diese Systeme werden durch das von menschlichen Experten abgefragte Wissen konstruiert. Dieses Expertenwissen wird in einer Form codiert, die einem Computer die Verwendung zur Bewertung alternativer Lösungen ermöglicht [Aronson, 2001].

Beratungssysteme helfen bei der Lösung von Problemen, die normalerweise durch menschliche Experten gelöst werden. Als solche können Beratungssysteme als eine Art Expertensystem klassifiziert werden [Forslund, 1995].

Während Beratungssysteme und Expertensysteme eine ähnliche Architektur besitzen, unterscheiden sie sich in einigen wesentlichen Punkten. Expertensysteme sind in der Regel eigenständige Problemlösungssysteme für Situationen, in denen das Problem klar definiert ist und das Expertenwissen angewendet werden kann, um die passende Lösung zu finden. Im Gegensatz dazu treffen Beratungssysteme keine Entscheidungen, sondern führen die Entscheidungsträger in ihrem Entscheidungsprozess, während die endgültige Entscheidung weiterhin durch den menschlichen Benutzer getroffen wird. Der menschliche Entscheider arbeitet mit dem

Beratungssystem, um Probleme zu identifizieren, die angegangen werden müssen und für diese iterativ die möglichen Lösungen zu bewerten [Aronson, 2001].

2.2.5 Geographische Informationssysteme

Die traditionelle Darstellung von geographischen Informationen sind Karten. Diese werden bereits seit Tausenden von Jahren benutzt. Eine individuelle Karte enthält viele Informationen, die von verschiedenen Personen und Organisationen verschiedenartig benutzt werden. Sie repräsentiert die Bedeutung von Orten in der Welt um uns. Karten werden für die unterschiedlichsten Anwendungen benutzt, von der Übersicht von verlegten Telefonkabeln und Gasrohren unter der Straße bis zur Abholzung des Regenwaldes im brasilianischen Amazonasgebiet.

*Definition 17: Ein **Geoinformationssystem** (GIS, engl. Geographical Information System) dient der Erfassung, Speicherung, Analyse und Darstellung aller Daten, die einen Teil der Erdoberfläche und die darauf befindlichen technischen und administrativen Einrichtungen sowie geowissenschaftliche, ökonomische und ökologische Gegebenheiten beschreiben [Bartelme, 1989].*

Das Wort Geographisch (verkürzt zu Geo) impliziert, dass der Ort eines jeden Datenobjekts, auf den es sich bezieht (Raumbezug), in geographischen Koordinaten (Breiten- und Längengrad) bekannt ist oder berechnet werden kann. Die meisten GIS sind auf zwei räumliche Dimensionen beschränkt. Es gibt aber auch Systeme, die eine echte dreidimensionale Verarbeitung erlauben; mit ihnen lassen sich Objekte wie zum Beispiel mehrere Gesteinsschichten darstellen [Moßgraber, 1997].

Der Begriff der Geoinformation (engl. Geographical Information oder Spatial Information) ergibt sich durch Spezialisierung auf Information, die orts-

raum- und zeitbezogenen Charakter hat. Auf die gleiche Art und Weise ergibt sich der Begriff Geodaten (engl. Geographical Data) [Bartelme, 2000].

2.2.5.1 Aufgaben von GIS

Der übergreifende Verwendungszweck von GIS ist die Unterstützung von Entscheidungen, die auf räumlichen Daten basieren. Diese Aufgabe löst ein GIS durch Anwendung einer oder mehrerer der folgenden Techniken, die in Tabelle 3 näher beschrieben werden: Organisation, Visualisierung, Räumliche Suche, Kombination und Analyse [Bartelme, 2000].

Tabelle 3: Techniken von GIS

Organisation	Der wichtigste Punkt für die Organisation von GIS-Daten ist die räumliche Lage. Die Daten einer Untersuchung der Luftzusammensetzung hätte ohne einen <i>Raumbezug</i> keinerlei Aussagekraft. GIS-Daten werden aber auch nach Merkmalen organisiert, die sich nicht auf einen Raum beziehen, zum Beispiel könnte eine Luftuntersuchung jedes Jahr durchgeführt und die Daten dann nach der Jahreszahl organisiert werden. Dies nennt man den <i>Zeitbezug</i> .
Visualisierung	Menschen haben die außergewöhnliche Fähigkeit, komplexe räumliche Beziehungen graphisch erkennen zu können. Liegt dieselbe Information dagegen in Form einer Tabelle mit Zahlenwerten vor, ist sie meist völlig unverständlich. Die Visualisierung wird in GIS durch Farben, Symbole und spezielle Methoden, die verschiedene Perspektiven, Schattierungen und ähnliches verwenden, erreicht.

Räumliche Suche	Viele GIS benutzen eine Abfragesprache, um spezifische Fragen zu formulieren. Diese Anfragen können sehr komplex ausfallen, zum Beispiel: "Ermittle alle Orte im Umkreis von 250 m um einen See, die 1000 m über dem Meeresspiegel liegen und an denen der Untergrund aus Kalkstein besteht." Andere Arten von Anfragen können sich auf die Lage von Objekten zueinander beziehen (zum Beispiel Entfernungen, Orientierungen, Nachbarschaftsbeziehungen oder noch komplexere Dinge).
Kombination	Die Möglichkeit, verschiedene räumliche Datensätze aus unterschiedlichen Informationsquellen zu kombinieren und anzuzeigen, dient oft dem Verständnis und führt zu Interpretationen von räumlichen Phänomenen, die bei separater Betrachtung nicht sichtbar sind. Der Prozess des Kombinierens von Schichten räumlicher Daten wird meist als Datenintegration oder <i>Verschneidung</i> bezeichnet und wird ausgeführt durch das visuelle Übereinanderlegen von Grafiken verschiedenster Art oder durch Integrationsmodelle, die aus mehreren Karten und Tabellen mit Attributen eine neue Karte erzeugen. Ein Hauptproblem bei Verschneidungen ist, dass die Raumbezugssysteme der unterschiedlichen Karten und Tabellen meist nicht übereinstimmen und erst angepasst werden müssen.
Analyse	Durch den Prozess der Analyse möchte man aus Daten Informationen gewinnen. Die Analyse wird in GIS meist visuell ausgeführt, kann aber auch durch Messungen, statistische Auswertungen und andere Operationen erfolgen.

2.2.5.2 Standards für Geoinformationssysteme

Die wichtigsten Standards im GIS-Bereich sind die Standards des *Open Geospatial Consortiums* (OGC) und die ISO Serie 191xx. Das OGC ist eine 1994 als *Open GIS Consortium* gegründete gemeinnützige Organisation. Das Konsortium hat mehr als 400 Mitglieder aus dem Kreis von Regierungsorganisationen, Firmen und Universitäten. Das OGC folgt einem Konsensprozess bei der Entwicklung von Spezifikationen und Standards für die Interoperabilität, den Austausch und die Bereitstellung geografischer Daten und Dienste. Hauptziel ist es, den Zugang zu geografischen Diensten und

geografischen Daten über das Internet zu ermöglichen. Das OGC seinerseits ist seit Januar 2007 Mitglied des World Wide Web Consortiums (W3C).

OGC Sensor Web Enablement

Das *Sensor Web Enablement* (SWE) ist eine Initiative des OGC mit dem Ziel, ein Rahmenwerk offener Standards für den Zugriff auf alle Arten von Sensordaten über das Internet zu definieren. Diese Sensordaten können aus einem Archiv stammen oder direkt aus einem über ein Netzwerk zugänglichen Sensornetz abgerufen werden [Botts, 2008]. Tabelle 4 beschreibt die Standards von OGC SWE.

Tabelle 4: OGC SWE Standards

Sensor Observation Service (SOS) ⁴	SOS ist eine Schnittstellenspezifikation zur Abfrage von (Echtzeit-) Sensordaten sowie von Sensordatenzeitreihen. Die angebotenen Sensordaten umfassen einerseits Beschreibungen von Sensoren selbst, die in der Sensor Model Language (SensorML) kodiert werden sowie die Messwerte, die in dem Observations & Measurements (O&M) Format kodiert werden. Falls der SOS transaktional (SOS-T) ist, können auch neue Messwerte eingefügt werden.
Sensor Planning Service (SPS) ⁵	Der SPS Standard definiert Schnittstellen für Abfragen, die Informationen über die Fähigkeiten eines Sensors zur Verfügung stellen und wie der Sensor ansprechbar ist.

⁴ <http://www.opengeospatial.org/standards/sos>, letzter Abruf am 4.9.2016

⁵ <http://www.opengeospatial.org/standards/sps>, letzter Abruf am 4.9.2016

Sensor Model Language (SensorML) ⁶	SensorML beschreibt ein Informationsmodell, welches das Auffinden und Nutzbarmachen von Web-verfügbaren Sensoren ermöglicht. Innerhalb dieses Kontextes erlaubt SensorML dem Entwickler, Modelle und Schemata zu definieren, sowie jeglichen Prozess inklusive der Messungen eines Sensorsystems zu beschreiben und zu verarbeiten.
Observations & Measurements (O&M) ⁷	Die O&M-Spezifikation enthält allgemeine Modelle und Codierungen für Sensorbeobachtungen und Messungen. Sie wird in allen anderen SWE Standards als Datenmodell verwendet. Für O&M sind bereits mehrere Übersetzungen in eine Ontologie verfügbar, beispielsweise die Semantic Sensor Network Ontology (SSNO) [Compton, 2012].
Sensor Event Service (SES)	Der SES ist eine Erweiterung des Sensor Alert Service (SAS). Beide werden verwendet, um eine Publish & Subscribe basierte Zugriffsart auf Sensormessungen zu realisieren. Sie bieten auch optionale Methoden, um dynamisch neue Sensoren hinzuzufügen und Benachrichtigungen an den Dienst senden.
SensorThings API ⁸	Die SensorThings API (Application Programming Interface ⁹) entstand mit Bezug auf das Internet der Dinge. Das Datenmodell basiert auf O&M, sodass es mit SOS-Diensten zusammenarbeiten kann. Der wesentliche Unterschied besteht darin, dass die SensorThings API leichtgewichtiger Technologie wie Representational State Transfer (REST) ¹⁰ einsetzt.

⁶ <http://www.opengeospatial.org/standards/sensorml>, letzter Abruf am 4.9.2016

⁷ <http://www.opengeospatial.org/standards/om>, letzter Abruf am 4.9.2016

⁸ <http://www.opengis.net/doc/is/sensorthings/1.0>, letzter Abruf am 4.9.2016

⁹ Programmierschnittstelle

¹⁰ REST bezeichnet ein Programmierparadigma für verteilte Systeme und ist eine Abstraktion der Struktur und des Verhaltens des World Wide Web.

Tabelle 5 beschreibt weitere für diese Arbeit relevante OGC Standards.

Tabelle 5: Weitere OGC Standards

Catalogue Services (CAT) ¹¹	CATs ermöglichen die Veröffentlichung und Suche von beschreibenden Informationen (Metadaten) für Daten, Dienste und die damit verbundenen Informationsobjekte.
Catalogue Services for the Web (CSW)	Metadaten in Katalogen stellen Ressourceneigenschaften dar, die zur Auswertung und weiteren Verarbeitung abgefragt und dargestellt werden können. Die Bindung der Schnittstelle an das Hypertext Transfer Protocol (HTTP) ¹² wird als CSW bezeichnet und ist Teil der Spezifikation.
Geography Markup Language (GML) ¹³	GML erlaubt die Übermittlung von Geodaten mit Attributen, Relationen und Geometrien unter Einbeziehung von nicht-konventionellen Daten wie Sensordaten.
Web Feature Service (WFS) ¹⁴	Der WFS ermöglicht den Zugriff auf geographische Merkmale (engl. Feature) in Datenbanken und gibt das Ergebnis in GML zurück, wobei auch zusätzliche Formate unterstützt werden können. Unter einem Merkmal versteht man die allgemeine Abstraktion eines realen Faktums (engl. Real world phenomenon).
Web Map Service (WMS) ¹⁵	WMS ist ein Standardprotokoll zur Abfrage georeferenzierter Kartenbilder(-ausschnitte), die aus Daten die in einer GIS-Datenbank gespeichert sind, generiert werden.

¹¹ <http://www.opengeospatial.org/standards/cat>, letzter Abruf am 4.9.2016

¹² <https://www.w3.org/Protocols/rfc2616/rfc2616.txt>, letzter Abruf am 4.9.2016

¹³ <http://www.opengeospatial.org/standards/gml>, letzter Abruf am 4.9.2016

¹⁴ <http://www.opengeospatial.org/standards/wfs>, letzter Abruf am 4.9.2016

¹⁵ <http://www.opengeospatial.org/standards/wms>, letzter Abruf am 4.9.2016

Web Processing Service (WPS) ¹⁶	Die WPS Spezifikation enthält Regeln, um Ein- und Ausgänge (Anfragen und Antworten) für GIS-Verarbeitungsdienste beschreiben zu können. Der Standard definiert auch, wie ein Client die Ausführung eines Prozesses anfordern kann und wie auf die Ausgabe der Berechnung zugegriffen werden kann, etwa über Netzwerk-Streaming oder zum Download auf einem Server.
Web Notification Service (WNS) ¹⁷	Web-Service-Schnittstelle für die asynchrone Übermittlung von Nachrichten oder Warnungen. Diese Spezifikation liegt bereits seit 2006 als Entwurf vor, wird aber weiterhin als zentrales Element von SWE geführt.
WaterML ¹⁸	WaterML dient zur Enkodierung hydrologischer Zeitreihen und ist ein standardisiertes Informationsmodell für die Darstellung von Wassermesswerten, das den Datenaustausch zwischen Informationssystemen ermöglicht.

ISO Serie 191xx

Das OGC arbeitet eng zusammen mit dem Technischen Komitee 211 (TC211) der ISO, das für die ISO-19100-Normenreihe verantwortlich ist. Die ISO 19100-Serie definiert einige wichtige Grundlagen im Bereich der geografischen Informationen und Geoinformatik¹⁹. Mehrere OGC Standards (zum Beispiel GML) sind in der 191xx Reihe als ISO Standard aufgenommen worden. Eine vollständige Übersicht aller zugehörigen Standards findet man auf der Webseite des TC211²⁰.

¹⁶ <http://www.openeospatial.org/standards/wps>, letzter Abruf am 4.9.2016

¹⁷ http://portal.openeospatial.org/files/?artifact_id=1367, letzter Abruf am 4.9.2016

¹⁸ <http://www.openeospatial.org/standards/waterml>, letzter Abruf am 4.9.2016

¹⁹ <http://www.openeospatial.org/>, letzter Abruf am 4.9.2016

²⁰ <http://www.isotc211.org/Outreach/Overview/Overview.htm>, letzter Abruf am 4.9.2016

2.2.6 Frühwarnsysteme

Frühwarnsysteme kommen deshalb in den unterschiedlichsten Bereichen vor, in denen eine Notwendigkeit zur Gefahrenwarnung besteht, wie beispielsweise bei durch die Natur oder den Menschen verursachten Katastrophen. Die Vorhersage basiert auf naturwissenschaftlich-technischen Ansätzen und ihr Ziel ist die möglichst genaue Angabe von Größe, Eintrittszeit, Zeitdauer des Prozesses und räumlicher Ausbreitung des Naturereignisses.

*Definition 18: **Frühwarnsysteme** (EWS, engl. Early Warning System) dienen zur möglichst frühzeitigen Information über eine sich anbahnende oder auftretende Gefahr, um Personen und Organisationen die Möglichkeit zu geben, entsprechend darauf reagieren zu können [UNISDR, 2009].*

In der Definition wird die *möglichst frühzeitige* Warnung gefordert. Außerdem wird im Zusammenhang mit Frühwarnsystemen häufig von *Echtzeitfähigkeit* gesprochen (beispielsweise zum Abruf der Messwerte und Verarbeitungsalgorithmen). In Tabelle 6 werden die Vorlaufzeiten, die notwendig sind, bis zur Auslieferung der Warnung für die unterschiedlichen Arten von Gefahren nach dem aktuellen Stand der Technik zusammengestellt. Anwendungsfelder für Frühwarnsysteme gibt es aber auch in der Industrie [Mayr, 2009], wenn zum Beispiel über einen drohenden Produktionsausfall wegen fehlender oder fehlerhafter Teile informiert werden soll oder an der Börse bei drohenden Kursverlusten [Pettway, 1980].

Tabelle 6: Vorlaufzeiten für die Frühwarnung [IFRC, 2012]

Sekunden bis Minuten	Stunden bis Tage	Wochen bis Monate	Jahre
Erdbeben	Starke Stürme	Dürren	Steigender Meeresspiegel
Industrieunfall	Flächenbrand	Extreme Temperaturen	Waldrodung
Staubteufel	Wirbelstürme		Wüstenausbreitung
Tornados	Erdrutsche		Trockenperioden
Sturzfluten	Überschwemmungen		Extreme Niederschläge
	Tsunamis		Bodendegradation
	Vulkanausbrüche		Umweltverschmutzung
	Hitzewellen		
	Epidemien		

2.3 Systemarchitektur

Das Wort *Systemarchitektur* setzt sich zusammen aus den Wörtern *System* und *Architektur*. Auf den Begriff des Systems wurde bereits im Abschnitt 2.2 eingegangen.

2.3.1 Architektur

Der Begriff der *Architektur* stellt eine eigene Fachdisziplin dar und stammt nicht aus der Informatik. In Anlehnung an diese klassische Architektur, die sich mit dem Aufbau und der Struktur von Gebäuden beschäftigt, wird in der Informatik der Begriff zur Strukturierung und dem Aufbau von Hard- und Software verwendet.

2.3.2 Definition des Systemarchitekturbegriffs

Die ISO definiert im *Open Distributed Processing - Reference Model* [ISO, 1995] die Systemarchitektur als

„... a set of rules to define the structure of the system and the interrelationships between its parts.“

Eine Architektur stellt also einer Menge von Regeln dar, die die Struktur des Systems beschreiben und die Beziehungen seiner Elemente. Die Definition einer Architektur, die im Standard des American National Standards Institute (ANSI) und Institute of Electrical and Electronics Engineers (IEEE) 1471-2000 verwendet wird, erweitert die ISO Definition um zusätzliche Aspekte [IEEE, 2000]:

„The fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution.“

Während hier auch auf die Organisation des Systems und die Beziehungen seiner Elemente Wert gelegt wird, kommen nun zusätzlich die Interaktion des Systems mit seiner Umgebung und der Aspekt der Architekturevolution vor. An der Aufnahme des Evolutionsbegriffs kann man das Umdenken in der Disziplin der Systemarchitektur erkennen: Architekturen sind nicht mehr *in Stein gemeißelt*, sondern müssen sich auch neuen Anforderungen stellen können und entsprechend flexibel anpassbar sein [Zimmermann, 2003].

Folgende Definition stellt eine Fusion aus obigen Definitionen dar. Der Evolutionsaspekt wurde jedoch nicht aufgenommen, da er in dieser Arbeit als Qualitätsmerkmal einer Architektur und nicht als integraler Bestandteil der Definition angesehen wird. Unter Verwendung der in der Definition des Systembegriffs verwendeten Begrifflichkeiten (s. Abschnitt 2.2) ergibt sich daraus folgende Definition:

Definition 19: *Eine **Systemarchitektur** besteht aus einer Menge von Regeln, welche die Struktur des Systems und seine Interaktion mit der Umgebung definieren sowie der Komposition seiner Elemente.*

2.3.3 Architekturmuster

Gamma, Helm, Johnson und Vlissides (besser bekannt als die *Gang of Four*²¹) definieren in der deutschen Ausgabe ihres Buchs *Design Patterns* [Gamma, 1996] ein Entwurfsmuster wie folgt:

*Definition 20: Ein **Entwurfsmuster** (engl. Design Pattern) benennt, erläutert und bewertet systematisch einen wichtigen und wiederkehrenden Entwurf in objektorientierten Systemen [Gamma, 1996].*

Weiterhin führen sie folgende Vorteile an, die den Einsatz von Entwurfsmustern nahelegen [Gamma, 1996]:

- Entwurfsmuster vereinfachen die Wiederverwendung von erfolgreichen Entwürfen und Architekturen.
- Die Darstellung bewährter Techniken als Entwurfsmuster macht die Techniken leichter verständlich, sodass Entwickler neuer Systeme einfacher auf sie zurückgreifen können.
- Entwurfsmuster helfen zwischen Entwurfsalternativen zu wählen.
- Entwurfsmuster können die Dokumentation und Wartung existierender Systeme verbessern, weil sie Klassen- und Objektinteraktionen explizit spezifizieren und den ihnen zugrundeliegenden Zweck erklären.

Im Gegensatz dazu werden in *Anti-Patterns* häufig anzutreffende schlechte Lösungsansätze für ein bestimmtes Problem dokumentiert mit dem Ziel, deren Verwendung nicht zu wiederholen.

²¹ https://en.wikipedia.org/wiki/Design_Patterns, letzter Abruf am 4.9.2016

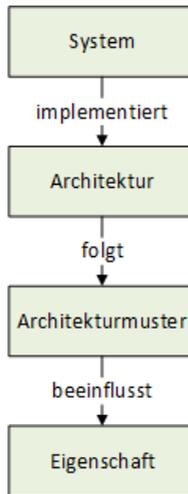


Abbildung 14: System, Architektur, Architekturmuster und Eigenschaft nach [Richling, 2011]

Richling [Richling, 2011] zeigt den Zusammenhang der Begriffe System, Systemeigenschaften, Architektur und Architekturmuster auf, wie in Abbildung 14 dargestellt. Ein System implementiert eine Architektur. Eine Architektur folgt wiederum einem oder mehreren Architekturmustern und ein Architekturmuster ermöglicht oder vermindert schließlich bestimmte Eigenschaften des Systems. Die Nutzung von Architekturmustern zur Umsetzung und Sicherstellung von Systemeigenschaften wird bei dieser Sichtweise als unverzichtbar angesehen.

2.3.4 Bewertung von Architekturen

Die Bewertung einer Softwarearchitektur stellt ein Instrument des Risikomanagements dar. Sie ermöglicht die frühzeitige Identifikation von Abweichungen in Bezug auf die Erreichung von Qualitätsanforderungen und die Zeit- und Budgetplanung. Das Verständnis der Auswirkungen der Entwurfsentscheidungen auf die Qualitätsmerkmale durch systematische

Analyse ist eine Voraussetzung für die Bewertung von Softwarearchitekturen. Der Aufwand für die Bewertung ist dann geringer und es ergibt sich ein gutes Kosten-/Nutzenverhältnis [Clements, 2002].

Die relativ große Anzahl verfügbarer Bewertungsverfahren mit ihren individuellen Eigenschaften erschwert die Auswahl des für ein konkretes Projekt passenden Verfahrens. Eicker, Hegmanns und Malich erstellten zur Entscheidungsunterstützung einen strukturierten Bewertungsprozess [Eicker, 2007]. Er basiert auf Entscheidungsbäumen, deren Blätter jeweils eine konkrete Methode oder eine Kombination aus Methoden hinterlegt und somit empfohlen ist. Als Einstiegspunkte in die Bäume dienen jeweils die gewünschten Qualitätskriterien und die Projektphasen, in denen die Methode eingesetzt werden soll. Im Gegensatz zur Messung von Quellcode anhand bestimmter Metriken liefert die Bewertung von Softwarearchitekturen keine Zahlen sondern qualitative Aussagen, inwieweit sie die Erreichung bestimmter Qualitätsmerkmale ermöglichen. Dabei gilt es zu ermitteln, welche spezifischen Kriterien die Architektur bzgl. ihrer *Güte* oder *Eignung* zu erfüllen hat. Starke stellt hierzu folgende drei Kriterien auf [Starke, 2008]:

- Das System (das mit dieser Architektur entwickelt wird oder wurde) erfüllt sowohl seine funktionalen als auch nichtfunktionalen Anforderungen (Qualitätskriterien).
- Insbesondere erfüllt das System die spezifischen Anforderungen an Flexibilität, Änderbarkeit und Performanceverhalten.
- Das System kann mit den zur Verfügung stehenden Ressourcen realisiert werden. Sowohl das Budget wie auch der Zeitplan, das Team, die beteiligten Fremd- und Legacy-Systeme, die vorhandene Basistechnologie und Infrastruktur sowie die gesamte Organisationsstruktur beeinflussen die Umsetzbarkeit einer Architektur.

Die aufgestellten Qualitätsmerkmale sollten durch Szenarien für die wichtigsten Qualitäts- und Architekturziele verfeinert werden. Diese werden

gemeinsam mit den maßgeblichen Beteiligten (engl. Stakeholder) definiert. Abbildung 15 zeigt beispielhaft wie ein Qualitätsbaum der aufgestellten Qualitätskriterien und zugeordnete Szenarien aussehen kann.

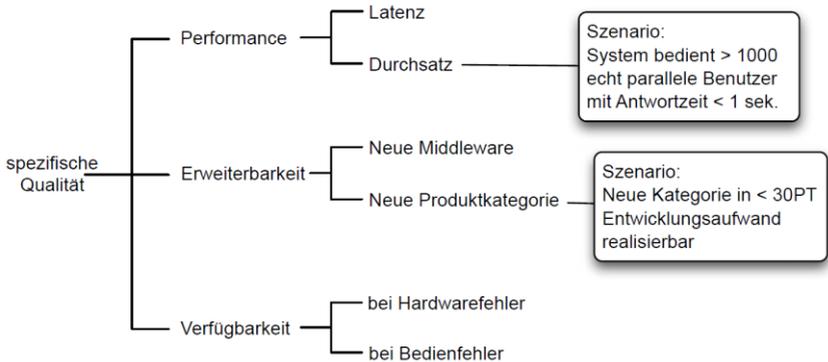


Abbildung 15: Qualitätsbaum mit Szenarien [Starke, 2008]

Auditoren einer Architektur untersuchen gemeinsam mit den Architekten des Systems in Workshops die zentralen Architektur- und Qualitätsziele auf Basis der jeweils definierten Szenarien.

2.4 Beschreibung von Systemarchitektur

Die Beschreibung einer Systemarchitektur gliedert sich in Modellierungssprachen und Rahmenwerke (s. Abbildung 16).

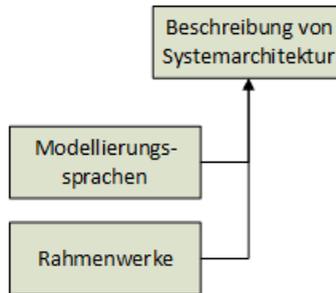


Abbildung 16: Themenbereich der Architekturbeschreibung

Die Open Group definiert in *The Open Group Architecture Framework* (TOGAF) [TOG, 2011] den Begriff *Architekturbeschreibung* folgendermaßen:

*Definition 21: Eine **Architekturbeschreibung** ist eine formale Beschreibung eines Informationssystems, die in einer bestimmten Art und Weise organisiert ist, sodass sie die Argumentation über die strukturellen Eigenschaften des Systems unterstützt. Die Beschreibung definiert die Komponenten oder Bausteine, die das gesamte Informationssystem bilden. Sie beinhaltet weiterhin einen Plan, der beschreibt, wie Teilsysteme entwickelt oder beschafft werden können, um das Gesamtsystem zu implementieren [OpenGroup, 1995].*

Zur Architekturbeschreibung können eine oder mehrere Modellierungssprachen eingesetzt werden.

2.4.1 Modellierungssprachen

Modellierungssprachen ermöglichen Softwareentwicklern, Systemanalytikern oder Systemarchitekten, die Anforderungen an ein Organisations- oder Softwaresystem sowie dessen Strukturen und inneren Abläufe auf einer höheren Abstraktionsebene festzulegen. Diese Sprachen versuchen

eine Spezifikation für das Management, Benutzer und andere Beteiligte durch Darstellung in Diagrammform möglichst verständlich zu machen. Jede Modellierungssprache verfügt über eine festgelegte Syntax, die über eine Grammatik oder ein Metamodell beschrieben werden kann [Hesse, 2008]. Eine Modellierungssprache kann grafisch oder textuell sein [He, 2007]:

- *Grafische Modellierungssprachen* verwenden Diagrammtechniken. Ein Diagramm enthält benannte Symbole, die Konzepte repräsentieren. Linien verbinden die Symbole, um Beziehungen und Einschränkungen zwischen den Symbolen darzustellen.
- *Textuelle Modellierungssprachen* verwenden standardisierte Schlagworte, die zusätzlich von Parametern oder natürlichsprachlichen Begriffen und Phrasen begleitet werden können, um Computer-interpretierbare Ausdrücke zu erstellen.

In den folgenden Abschnitten werden zunächst vielfältig einsetzbare Modellierungssprachen wie die Unified Modeling Language (UML) vorgestellt, danach Ontologien zur semantischen Wissensrepräsentation und abschließend Sprachen zur Beschreibung von Prozessen und Regeln.

2.4.1.1 Allgemeine Modellierungssprachen

Unified Modeling Language

UML ist eine grafische Modellierungssprache zur Spezifikation, Konstruktion und Dokumentation von Software-Teilen und anderen Systemen [Booch, 1998]. Sie wurde von der Object Management Group (OMG)²² entwickelt und ist für die aktuelle Version 2.4.1 als ISO/IEC 19505 standardisiert.

Die Diagramme von UML 2 lassen sich in zwei Kategorien einteilen: Strukturdiagramme und Verhaltensdiagramme. Auf die zahlreichen Diagrammtypen und ihre Ausdrucksmöglichkeiten wird in dieser Arbeit nicht näher eingegangen. Der interessierte Leser findet eine Einführung beispielsweise in *UML 2 glasklar* [Jeckle, 2004].

Fundamental Modeling Concepts

Die Firma SAP entwickelte Fundamental Modeling Concepts (FMC), da sie diverse Kritikpunkte am Industriestandard UML hatte. Ihr Hauptkritikpunkt war, dass die Komponentenbeziehungen keine große Aussagekraft haben und die Kommunikation und der Informationsaustausch mehr in den Vordergrund rücken müssten [Knöpfel, 2006]. FMC soll es deshalb Menschen ermöglichen, die Konzepte und Strukturen komplexer Informationssysteme auf effiziente Art und Weise zwischen den verschiedenen Arten von Interessengruppen zu kommunizieren. Die Notation ist aus bestehenden Standards entliehen und ist leicht zu erlernen, um die Strukturen von Informationssystemen zu visualisieren und in kohärenter Weise zu kommunizieren. Auf allen Abstraktionsebenen wird versucht, konzeptionelle Strukturen von Führungsstrukturen zu trennen. Agenten werden als Rechtecke dargestellt, während Standorte als abgerundete Rechtecke symbolisiert sind. Kommunikationskanäle werden als kleine Kreise und Datenspeicher als größere Kreise oder abgerundeten

²² <http://www.omg.org/>, letzter Abruf am 4.9.2016

Rechtecke dargestellt. Gebogene Pfeile symbolisieren, ob ein Agent Informationen von einem Speicher liest oder schreibt.

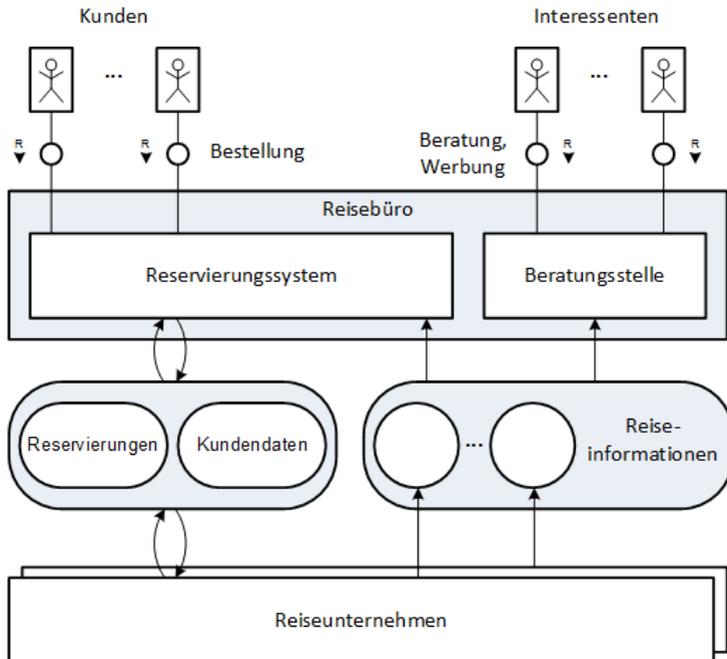


Abbildung 17: Beispiel für FMC Blockdiagramm²³

In Abbildung 17 wird ein Beispiel für ein FMC Blockdiagramm gezeigt. Auffällig im Gegensatz zu einem UML Diagramm sind die Verbindungslinien mit kleinen Kreisen über die, die Akteure verbunden sind. Diese können optional eine Richtung beinhalten (kleiner Pfeil), um den Urheber der Kommunikation zu verdeutlichen. Fehlt dieser, ist die Kommunikation in beide Richtungen. Die großen Kreise stellen Datenspeicher dar. Die auf sie

²³ S. <http://www.fmc-modeling.org/quick-intro>, letzter Abruf am 4.9.2016

zu- und von ihnen wegführenden Pfeile stellen die Schreib- bzw. Lesezugriffe dar.

2.4.1.2 Ontologien

Der Begriff Ontologie stammt aus der theoretischen Philosophie und bezeichnet dort in der deutschen Bedeutung *die Lehre vom Sein, vom Seienden* [Duden, 2011]. In der Informatik stellen Ontologien ein beliebtes Mittel zur Strukturierung, Modellierung und Repräsentation von Wissen dar. Insbesondere auch im Rahmen des *Semantic Webs* finden Ontologien zunehmend Einsatz in Unternehmen, um bestehende Daten semantisch zugänglich zu machen. Ziel des *Semantic Webs* ist es, die Bedeutung von Begriffen für einen Computer auswertbar zu machen. Aufgrund der vielfältigen Einsatzmöglichkeiten von Ontologien ist es schwierig, eine klare Definition für den Begriff Ontologie zu finden. Das *Handbook on Ontologies* [Staab, 2004] definiert eine Ontologie als

„A formal explicit specification of a shared conceptualization for a domain of interest“,

also eine formale, explizite Spezifikation einer gemeinsam genutzten Konzeptualisierung. Stuckenschmidt kritisiert diese Definition, weil sie die meisten aus der Linguistik stammenden, im mathematischen Sinne nicht formalen Ansätze, ausschließt [Stuckenschmidt, 2009]. Eine Ontologie, die nur von einer Person genutzt wird, ist nach der Definition von Staab und Studer keine Ontologie mehr, weil sie eben das Kriterium der gemeinsamen Nutzung nicht erfüllt [Staab, 2004]. Deshalb soll im Folgenden die meistverwendete Definition des Begriffs von Gruber gelten [Gruber, 1995]:

*Definition 22: Eine **Ontologie** ist eine explizite Spezifikation einer Konzeptualisierung.*

Im Kontext semantischer Technologien wird der Begriff Wissensbasis (engl. Knowledge Base) als äquivalent zum Begriff Ontologie verstanden. Beispielsweise verwendet auch Hitzler [Hitzler, 2008] die beiden Begriffe

synonym. In dieser Arbeit steht jedoch der Begriff Ontologie für eine Terminologie einer Anwendungsdomäne und der Begriff Wissensbasis für eine instanziierte Ontologie, also für eine Ontologie mit einer Beschreibung der Welt.

Um eine Ontologie zu spezifizieren, wird eine Sprache mit einer formalen Semantik benötigt [Staab, 2004]. Erst hierdurch können Konzepte und Relationen für die Individuen (*das Unteilbare* [Duden, 2011]) einer Ontologie definiert werden. Individuen sind hierbei die elementaren Bestandteile einer Ontologie, sie können Eigenschaften haben und sind Instanzen eines Konzeptes, dienen aber selbst nicht der Konzeptualisierung. Einige Ontologiesprachen erlauben jedoch die Verwendung von Konzepten als Individuen. Semantische Eigenschaften geben einem Objekt seine Bedeutung und beschreiben seine Beziehungen zu anderen Objekten auf einer abstrakten Ebene im Gegensatz zu physischen und messbaren Eigenschaften. Zum Beispiel fügt die Rolle einer Person in einem Geschäft (Kunde, Mitarbeiter, Manager, Berater, etc.) dieser Person eine neue Bedeutung hinzu. Eine Beziehungseigenschaft zwischen zwei Personen kann beispielsweise die Aussage sein, dass Herr Schmitt ein Kunde von Herrn Klaus ist. Die Spezifikation der semantischen Eigenschaften ist vom Anwendungskontext abhängig. Daher sollten sie von vorgegebenen Anwendungsszenarien abgeleitet werden.

Des Weiteren ist nach Staab [Staab, 2004] die Spezifikation einer Ontologie auch immer ein sozialer Prozess, da sich die beteiligten Menschen auf eine Konzeptualisierung einigen müssen. Deshalb ist eine Ontologie auch immer nur für eine eingeschränkte Wissensdomäne einsetzbar.

Auf der Basis der formalen Semantik einer Ontologie lassen sich zusätzlich logische Regeln definieren, die das Schlussfolgern von zusätzlichen Aussagen aus dem vorhandenen Wissen ermöglichen. Dies ist einer der größten Vorteile, die Ontologien gegenüber anderen Formen der Wissensrepräsentation bieten. Logische Regeln sind daher auch Bestandteil

der Semantik von Ontologiesprachen. Dieses Verfahren zum Folgern von logischen Schlüssen wird mit dem englischen Begriff *Reasoning* bezeichnet.

Ein weiterer Punkt, der Ontologien von anderen Wissensrepräsentationen unterscheidet, ist die sogenannte *Annahme zur Weltoffenheit* (OWA, engl. Open World Assumption). Sie besagt, dass neben den in der Ontologie vorkommenden Individuen weitere Individuen existieren können, sofern diese nicht explizit ausgeschlossen wurden. Im Gegensatz dazu wird bei der *Annahme zur Weltabgeschlossenheit* (CWA, engl. Closed World Assumption) angenommen, dass in der Wissensbasis alle Individuen enthalten sind.

Klassifikation von Ontologien

Nach ihrem Umfang lassen sich die Ontologien folgenderweise klassifizieren: (1) allgemeine, bereichsübergreifende (engl. Top level ontologies), (2) auf bestimmte Anwendungsbereiche bezogene (engl. Domain ontologies), (3) die speziellen Aspekte der Anwendung beschreibende Anwendungs- und Aufgabenontologien (engl. Application and Task ontologies) [Wand, 1990].

Die Ontologien der Kategorie (1) werden auch als Basisontologien (engl. Upper-level oder Foundational Ontologies) bezeichnet. Das in solchen Ontologien dargestellte Wissen ist allgemeiner Natur, beschreibt also im Sinne klassischer Categoriesysteme grundlegende Konzepte der Welt. Im Gegensatz dazu decken die Ontologien aus der Kategorie (2) ein spezielles Wissensgebiet ab. Anwendungsontologien aus der Kategorie (3) beschäftigen sich weniger mit einer neutralen Beschreibung eines Wissensgebietes, sondern mit speziellen Aspekten der Anwendung. Aufgabenontologien stellen dabei spezielle Anwendungsontologien dar, welche Definitionen enthalten, mit deren Hilfe eine gegebene Aufgabenstellung beschrieben werden kann [Stuckenschmidt, 2009].

Anwendungsfelder von Ontologien

Die Verwendung der Ontologien in der Informatik lässt sich nach Gruninger in drei Anwendungsfelder aufteilen [Gruninger, 2002]:

1. Kommunikation
2. automatisches Schließen und Repräsentation
3. Wiederverwendung von Wissen

Bei der Kommunikation handelt es sich um die Interpretationsvorschrift für die Daten in Form von semantischen Metadaten, deren Ontologie beiden Seiten zugänglich ist (beispielsweise zwei kommunizierende Programme). Beim automatischen Schließen wird das Ziehen von logischen Schlüssen aufgrund der aus der Ontologie bekannten Ableitungsregeln durchgeführt. Ähnlich ist es bei der Wissensrepräsentation und -wiederverwendung. Unter die Wissensrepräsentation fällt auch der wichtige Anwendungsbereich der Abbildung bzw. Integration unterschiedlicher Datenquellen (oder auch Funktionen), bei denen Ontologien häufig zum Einsatz kommen (engl. Information Integration). Hierzu bildet man die beiden Ontologien, die jeweils eine Datenquelle beschreiben, aufeinander ab. Man spricht in der Literatur dann von *Ontology Mapping* [Kalfoglou, 2003].

Semantische Annotation von Dokumenten

Der Prozess der Anreicherung von Dokumenten mit semantischen Metadaten sowie das Ergebnis dieses Prozesses wird als semantische Annotation bezeichnet. Semantische Metadaten sind Metadaten, welche mit Hilfe des in einer Ontologie spezifizierten Wissens generiert werden. Handschuh sieht semantische Annotationen von Web-Dokumenten als die wichtigste Komponente der Ansätze im Semantic Web [Handschuh, 2005].

Im Rahmen der semantischen Analyse müssen Konzepte, Relationen und deren Instanzen einer entsprechenden Ontologie im Text erkannt werden. Dies geschieht mit Hilfe einzelner Schritte des *Natural Language Processing*

(NLP). Semantische Annotationen können beispielsweise bei textbasierter Suche zur Erhöhung der Anzahl der gefundenen Ergebnisse (und folglich die Verbesserung der Recall-Werte) und zur Verbesserung der Reihung der Suchergebnisse beitragen [Iofciu, 2005].

Semantic Web

Das Semantic Web ist ein Konzept zur Verbesserung des aktuellen World Wide Web (WWW). Die Grundidee ist, die vorhandenen Daten so aufzubereiten, dass Computer diese verarbeiten und interpretieren können, um damit dem Menschen zu helfen, die gewünschten Informationen zu finden. Ähnlich wie das WWW ein großes verteiltes Hypertext-System ist, soll das Semantic Web ein großes verteiltes System zur Speicherung von Wissen bilden. Hauptsächlich wird die Idee durch das W3C vorangetrieben²⁴. Das Semantic Web baut, wie der Name bereits impliziert, auf semantischen Technologien auf. Die Architektur des Semantic Web zeigt Abbildung 18. Die jeweiligen Schichten des von Tim Berners-Lee erstellten Diagramms bauen aufeinander auf und sind zunächst technikneutral gehalten [Berners-Lee, 2000].

²⁴ <https://www.w3.org/standards/semanticweb/>, letzter Abruf am 4.9.2016

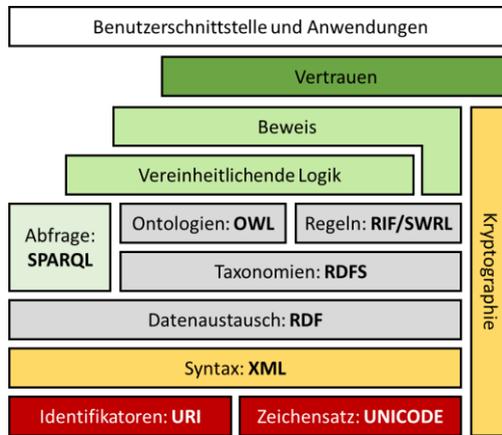


Abbildung 18: Der Semantic Web Stack nach [Berners-Lee, 2000]

Jedoch haben sich bereits für die meisten Schichten Standards herausgebildet und das W3C arbeitet weiterhin an der technologischen Ausgestaltung der fehlenden Schichten. Ähnliche Technologien für die Technologievorgaben des W3C wurden mittlerweile fast völlig verdrängt.

Die erste Basisschicht, bestehend aus Uniform Resource Identifier (URI)²⁵ und Unicode²⁶, folgt den Merkmalen des bestehenden WWW. Unicode ist ein Standard zur Codierung internationaler Zeichensätze und ermöglicht es, dass alle menschlichen Sprachen in den Dokumenten des WWW verwendet werden können. Ein URI ist eine Zeichenfolge in standardisierter Form, um Ressourcen eindeutig identifizieren zu können (Dokumente, Bilder, etc.). Eine Untergruppe des URI ist der Uniform Resource Locator (URL), der zusätzlich den gewünschten Zugriffsmechanismus (Netzwerkprotokoll) sowie die Position im Netzwerk enthält – wie beispielsweise <http://www.example.org/>. Eine internationale Variante der URI ist der

²⁵ <http://tools.ietf.org/html/rfc3986>, letzter Abruf am 4.9.2016

²⁶ <http://unicode.org/>, letzter Abruf am 4.9.2016

Internationalized Resource Identifier (IRI)²⁷, der die Verwendung von Unicode-Zeichen innerhalb einer URI ermöglicht. Wenn in dieser Arbeit von URIs gesprochen wird, kann man diese auch gedanklich durch IRIs ersetzen.

Für die Syntaxschicht wird die Extensible Markup Language (XML) verwendet. XML ist eine generische Auszeichnungssprache für strukturierte Dokumente. Zusammen mit den zugehörigen Standards XML-Namespaces²⁸ und XML-Schema²⁹ stellt XML sicher, dass eine gemeinsame Syntax im semantischen Web verwendet wird. Der Einsatz von Namensräumen (engl. Namespaces) ermöglicht, dass verschiedene Markup-Vokabulare in einem XML-Dokument parallel verwendet werden können. XML-Schema dient zur Definition der Struktur und deren Überprüfung eines bestimmten Satzes von XML-Dokumenten.

Ein elementares Darstellungsformat für das Semantic Web ist das *Resource Description Framework* (RDF)³⁰. RDF dient dazu, Informationen über Ressourcen (Metadaten) bereitzustellen. Beispiele für solche Metadaten sind der Titel, der Autor oder das Änderungsdatum einer Webseite. Ein einzelnes Metadatum besteht aus einem Tripel aus Subjekt-Prädikat-Objekt³¹. Ein Satz von RDF Tripeln bildet immer einen Graphen. Alle Daten im Semantic Web verwenden RDF als die primäre Darstellungssprache. Die normative Syntax für die RDF Serialisierung ist XML in der RDF/XML-Form.

Das *Resource Description Framework Schema* (RDFS)³² erweitert RDF um die Möglichkeit, terminologisches Wissen in Form von Klassen- und Eigenschaftshierarchien und deren Zusammenhängen zu spezifizieren. Für die Umsetzung einfacher hierarchischer Taxonomien ist RDFS ausreichend. Ausdrucksstärkere Ontologien können mit der Web Ontology Language

²⁷ <https://tools.ietf.org/html/rfc3987>, letzter Abruf am 4.9.2016

²⁸ <https://www.w3.org/TR/REC-xml-names/>, letzter Abruf am 4.9.2016

²⁹ <https://www.w3.org/TR/xmlschema-0/>, letzter Abruf am 4.9.2016

³⁰ <http://www.w3.org/TR/REC-rdf-syntax/>, letzter Abruf am 4.9.2016

³¹ Beispiel: „Karl May“ (Subjekt) „ist Autor von“ (Prädikat) „Winnetou I“ (Objekt)

³² <https://www.w3.org/TR/rdf-schema/>, letzter Abruf am 4.9.2016

(OWL)³³ erstellt werden. OWL ist syntaktisch ebenfalls in RDF eingebettet und stellt wie bereits RDFS ein zusätzliches standardisiertes Vokabular zur Verfügung. OWL kommt in drei Varianten - *OWL Lite* für einfache Taxonomien, *OWL DL* für die vollständige Ausdruckstärke der Prädikatenlogik (engl. Description Logic) und schließlich *OWL Full* für die maximale Ausdruckstärke und syntaktische Freiheit von RDF. Um Regeln basierend auf den Konstrukten von RDF(S) und OWL zur Verfügung zu stellen, dienen Regelsprachen. Bereits existierende Standards sind das Rule Interchange Format (RIF)³⁴ und die Semantic Web Rule Language (SWRL)³⁵.

Zur Abfrage von RDF(S)-Daten und OWL-Ontologien dient die Sprache Simple Protocol and RDF Query Language (SPARQL). SPARQL verwendet RDF-Tripel für die Anfrage, ist aber von der Syntax an die Structured Query Language (SQL) angelehnt, sodass die Anfragen für Nutzer klassischer relationaler Datenbanken durchaus vertraut aussehen. Wie der Name schon ausdrückt, stellt SPARQL nicht nur eine Abfragesprache dar, sondern spezifiziert auch ein Protokoll für den Zugriff auf RDF-Daten.

Die durchaus komplexeren oberen Schichten wie Beweisbarkeit, Vertrauen und das Querschnittsthema Kryptographie stehen weiterhin im Fokus der Forschung.

2.4.1.3 Datenmodellierung

Abfragesprachen

Eine Abfragesprache oder Anfragesprache (engl. Query Language) ist eine formale Sprache, die benutzt wird, um nach Informationen zu suchen [Reiner, 1991]. Das Ergebnis ist eine Untermenge des angefragten Datenpools. Dies wird auch als *Filterung* bezeichnet. Abfragesprachen stehen in enger Beziehung zu dem Modell, welches zur Speicherung der

³³ <https://www.w3.org/TR/owl-features/>, letzter Abruf am 4.9.2016

³⁴ <https://www.w3.org/standards/techs/rif>, letzter Abruf am 4.9.2016

³⁵ <https://www.w3.org/Submission/SWRL/>, letzter Abruf am 4.9.2016

Daten verwendet wird. Beispiele für Modelle und ihrer zugehörigen Sprachen sind [Lang, 1995]:

- relationale Modelle
- deduktive Modelle
- objektorientierte Modelle
- strukturierte Datenmodelle

In dieser Arbeit sind im Wesentlichen Sprachen für relationale Modelle (wegen ihrer weiten Verbreitung), deduktive Modelle (wegen ihrer Ausdrucksstärke) und strukturierte Datenmodelle (gute Unterstützung räumlicher Abfragen in WFS/SOS) relevant.

Relationale Datenbankmanagementsysteme

Bei relationalen Datenbankmanagementsystemen (DBMS) gelten die sogenannten ACID-Eigenschaften (Atomicity, Consistency, Isolation, Durability), die durch das Konzept der Transaktion garantiert werden. Nach dem erfolgreichen Abschluss einer Transaktion sind entweder alle in ihr vorgenommenen Änderungen gesichert oder es besteht weiterhin der vorherige Zustand. Alle in einer Transaktion durchgeführten Aktionen stellen also eine atomare Einheit dar. Die weiteren Eigenschaften von ACID sind die Konsistenz, die Isolation und die Dauerhaftigkeit. Nach einer erfolgreichen Transaktion werden demnach auch die Datenkonsistenz, die Unabhängigkeit von anderen Transaktionen und die dauerhafte Sicherung der Daten garantiert.

NoSQL Datenbankmanagementsysteme

Der Begriff *NoSQL* (im Sinne von Not only SQL) wurde Anfang 2009 von Johan Oskarsson für eine Konferenz über verteilte strukturierte Datenspeicher neu eingeführt. Die Veranstaltung trug den Titel *NOSQL 2009*³⁶ und war der Versuch einer gemeinsamen Begriffsfindung für die

³⁶ http://blog.sym-link.com/2009/05/12/nosql_2009.html, letzter Abruf am 4.9.2016

wachsende Zahl an nicht relationalen und verteilten Datenspeichersystemen. Die meisten der Datenspeichertechnologien für BigData fallen ebenfalls in diese Kategorie.

Meist geben NoSQL Datenbanken eine der ACID-Eigenschaften auf bzw. schwächen diese ab, um dafür jedoch andere Vorteile zu bieten. Insbesondere bei verteilten DBMS gilt das CAP-Theorem (s. Abschnitt 2.2.3.1). Häufig wird die Konsistenz durch das Konzept der *Eventual Consistency*³⁷ ersetzt. Dies bedeutet, dass das DBMS zwar die Konsistenz garantiert, aber nicht definiert, wie viel Zeit benötigt wird, um diese zu erreichen. In der Praxis bedeutet dies, dass, wenn ein Datensatz in einen Knoten geschrieben wird, es eine gewisse Zeit dauert, bis diese Änderung auch auf allen anderen Knoten des verteilten DBMS erscheint [ORACLE, 2012].

BigData

Der Begriff *BigData* steht für weit mehr als nur die Größe des Datenvolumens (engl. Volume), das behandelt werden muss. Weitere Aspekte sind die Datenvielfalt bzw. die Heterogenität der Daten (engl. Variety) und die Geschwindigkeit, in der die Daten bearbeitet (engl. Velocity) und analysiert werden müssen [BITKOM, 2014]. Man spricht deshalb auch von den „drei BigData Vs“.

Der erste Aspekt von BigData ist das Volumen der Daten. Die typische Menge an Daten, ab denen man von BigData spricht, liegt im Bereich von Terabytes, Petabytes, Exabytes, Zettabytes oder gar Yottabytes. Dieses Volumen nimmt sehr schnell zu: Im Zeitraum von 2000 bis 2002 wurden mehr Daten produziert als in den 40.000 Jahren zuvor. Seitdem hat sich die Zunahme der Datenmenge noch weiter erhöht. In der Zwischenzeit hat der Faktor des Wachstums der Daten das Wachstum der Verarbeitungskapazität, wie sie *Moore's Gesetz* beschreibt, überholt [BITKOM, 2014]. Der zweite Aspekt ist die Vielfalt der Daten: Die

³⁷ Auf Deutsch etwa: „Führt letztendlich zur Konsistenz“.

verwendeten Daten kommen heutzutage aus den unterschiedlichsten Quellen und sind daher heterogen. Es können unstrukturierte Daten, semi-strukturierte Daten und auch (traditionell) strukturierte Daten auftreten. Der dritte Aspekt ist die Geschwindigkeit: Daten werden nun mit solch hohen Raten erzeugt, dass die Grenzen der Speichersysteme, Transportnetze und sogar der Rechenleistung überschritten werden können. Schließlich kann BigData auch durch die auf diese Daten angewendete Analytik charakterisiert werden. Ein breites Spektrum an Techniken wird hierbei verwendet, beispielsweise Algorithmen aus Text- und Data-Mining oder Mustererkennung.

Eines der ersten BigData Probleme war die Web-Suche. Die Firma Google sah sich durch den rasanten Anstieg der Anzahl von Websites und die Herausforderung, bessere Suchergebnisse zu produzieren, mit der Problematik konfrontiert, dass herkömmliche Ansätze nicht mehr ausreichten. Ein weiteres typisches BigData Problem ist die Erzeugung von Empfehlungen auf der Grundlage von Kundenbestellungen, wie es beispielsweise von der Firma Amazon verwendet wird. Dies führte zu dem Schluss, dass die traditionellen Modelle und Algorithmen ihre Grenzen erreicht haben und neue Paradigmen und Techniken erfunden und erforscht werden müssen, mit denen die Herausforderungen von BigData bewältigt werden können. Die zwei wichtigsten neuen Techniken sind der MapReduce Algorithmus und *breite Tabellen* (engl. BigTable).

Der MapReduce Algorithmus

Der MapReduce Algorithmus gehört zur Familie der „teile und herrsche“ Algorithmen. D.h. das Problem wird gelöst, indem es in kleine Teilprobleme aufgespalten wird, die im optimalen Fall parallel bearbeitet werden können. Deren Ergebnisse werden schließlich zur Lösung kombiniert. Dieser Ansatz wurde weiterhin inspiriert durch die Funktionen *map* und *reduce* aus funktionalen Programmiersprachen. MapReduce als Programmierparadigma für große Datenmengen wurde erfunden von der Firma Google und zusammen mit einer gleichnamigen Implementierung veröffentlicht [Dean,

2004]. Im *map*-Schritt wird das Problem in kleinere Teilprobleme zerlegt und auf mehrere Arbeitsknoten (engl. Work nodes) verteilt. Dies kann ggf. rekursiv wiederholt werden. Die Arbeitsknoten verarbeiten dann das jeweilige Teilproblem. Im anschließenden *reduce* Schritt werden alle Ergebnisse der Teilprobleme vom Hauptknoten eingesammelt, um sie zur eigentlichen Antwort des Ursprungsproblems zusammenzufassen.

BigTable

BigTable stammt wie *MapReduce* ebenfalls von der Firma Google und liefert neben dem Konzept auch eine Implementierung gleichen Namens.

Die ursprüngliche Veröffentlichung benutzt für BigTable die folgende Definition [Chang, 2006]:

„A BigTable is a sparse, distributed, persistent multi-dimensional sorted map. The map is indexed by a row key, column key, and a timestamp; each value in the map is an un-interpreted array of bytes.“

Im Kontrast zu traditionellen Datenbankmanagementsystemen besteht also nicht die Notwendigkeit, dass alle Zeilen die gleichen Spalten verwenden. Jede Zeile kann ihren individuellen Satz an Spalten besitzen. Dies ermöglicht die effiziente Speicherung von heterogenen Daten.

Die Implementierung von BigTable wurde von Google als verteiltes Speichersystem für große strukturierte Datenmengen konzipiert und wird beispielsweise bei Google Analytics oder Google Earth eingesetzt.

2.4.1.4 (Geschäfts-) Prozessmodellierung

Bevor Sprachen zur Prozessmodellierung vorgestellt werden, muss zunächst dargelegt werden, worum es sich bei einem Geschäftsprozess und dem zugehörigen Prozessmanagement handelt. Die *European Association of*

*Business Process Management (EABPM)*³⁸ definiert in der deutschen Übersetzung ihres Referenzwerks *BPM Common Body of Knowledge* [EABPM, 2011] die Begriffe wie folgt:

*Definition 23: Als **Prozess** wird eine Reihe von festgelegten Tätigkeiten (Aktivitäten, Aufgaben) definiert, die von Menschen oder Maschinen ausgeführt werden, um ein oder mehrere Ziele zu erreichen [EABPM, 2011].*

*Definition 24: Die englische Bezeichnung Business Process Management (BPM) wird verwendet für das Geschäftsprozessmanagement oder auch einfach **Prozessmanagement**. BPM ist ein systematischer Ansatz, um sowohl automatisierte als auch nicht-automatisierte Prozesse zu erfassen, zu gestalten, auszuführen, zu dokumentieren, zu messen, zu überwachen und zu steuern und damit nachhaltig die mit der Unternehmensstrategie abgestimmten Ziele zu erreichen [EABPM, 2011].*

Wird zur Abarbeitung eines Prozesses IT eingesetzt, spricht man von Prozessautomatisierung. Hierzu dient ein Business Process Management-System (BPMS). Freund und Rücker stellen heraus, dass Prozessautomatisierung nicht zwangsläufig bedeutet, dass der gesamte Prozess vollautomatisch abläuft [Freund, 2010].

Laut Allweyer werden die Begriffe BPMS und Workflow Management-System (WFMS) nicht einheitlich verwendet [Allweyer, 2014]. Beide umfassen jedoch als zentrales Element eine Komponente zur Prozessausführung (engl. Process Engine). In manchen Veröffentlichungen steht die Abkürzung BPMS auch für *Business Process Management Suite*, um auszudrücken, dass die entsprechende Software noch zusätzliche Komponenten umfasst.

³⁸ <http://www.eabpm.org/>, letzter Abruf am 4.9.2016

*Definition 25: Die **Process Engine** steuert den Prozess, indem sie menschliche Prozessbeteiligte über anstehende Aufgaben informiert und das Ergebnis der Erledigung verarbeitet (Human Workflow Management) und indem sie interne und externe IT-Systeme über Schnittstellen aufruft (Service-Orchestrierung). Die Entscheidung darüber, welche Aufgaben oder Service-Aufrufe unter welchen Bedingungen stattfinden sollen oder nicht, trifft die Process Engine mithilfe des technischen Prozessmodells und des Ergebnisses einer Aufgabenerledigung oder eines Service-Aufrufes. Es ist also nicht zwangsweise so, dass der Ablauf eines automatisierten Prozesses nicht mehr durch die Prozessbeteiligten beeinflussbar wäre [Freund, 2010].*

Zur Erstellung des technischen Prozessmodells dienen spezielle Sprachen.

WS-Business Process Execution Language

Die WS-Business Process Execution Language (BPEL) ist eine XML-basierte Sprache für die Definition und Ausführung von Geschäftsprozessen. Jeder Prozess besteht aus einer oder mehreren Aktivitäten, die durch Web-Services realisiert werden. BPEL ist ein industrieller Standard der Organization for the Advancement of Structured Information Standards (OASIS)³⁹ und Teil der sogenannten WS-*-Spezifikationen. Bei WS-* handelt es sich um eine Klasse von Spezifikationen im Kontext von Web-Services. Ein Web-Service ist ein Dienst, der es einem elektronischen Gerät ermöglicht, mit einem anderen elektronischen Gerät über das WWW zu kommunizieren. In einem Web-Service werden Web-Technologien wie HTTP, die ursprünglich für die Mensch-zu-Maschine-Kommunikation entworfen wurden, für die Maschine-zu-Maschine-Kommunikation (M2M) verwendet.

³⁹ <http://www.oasis-open.org/>, letzter Abruf am 4.9.2016

An BPEL wurde vielfältige Kritik geäußert. Es verfügt beispielsweise über keine eigene Notation, was den Umgang für fachliche Experten erschwert. Damit wird der hauptsächliche Vorteil von BPMS, nämlich die Kommunikation zwischen IT- und Fachabteilung, zunichtegemacht. Weiterhin wird der Prozesscode durch die XML Sprachelemente sehr schnell unübersichtlich [Berntheisel, 2009].

Business Process Model and Notation

Business Process Model and Notation (BPMN) wurde von der *Object Management Group (OMG)* ⁴⁰ spezifiziert und ist eine grafische Spezifikationsprache in der Wirtschaftsinformatik und im Prozessmanagement. Sie stellt Symbole zur Verfügung, mit denen Fach-, Methoden- und Informatikspezialisten Geschäftsprozesse und Arbeitsabläufe modellieren und dokumentieren können. Hierzu definiert sie die Notation und Semantik von Kollaborations-, Prozess- und Choreografiediagrammen [OMG, 2013].

- **Prozess**

Generell besteht ein Prozess (engl. Process) aus einer Folge von Ablaufelementen. Er beginnt mit einem Startereignis, enthält einige zusammenhängende Aktivitäten und schließt mit einem Endereignis (s. Abbildung 19). In den folgenden Abschnitten werden wichtige Elemente von BPMN zur Modellierung eines Prozesses vorgestellt.



Abbildung 19: Ein einfacher Prozess

⁴⁰ <http://www.omg.org/>, letzter Abruf am 4.9.2016

- **Aktivität**

Eine Aktivität (engl. Activity) stellt einen Arbeitsschritt im Prozess dar. BPMN sieht mehrere Arten von Aktivitäten vor (s. Abbildung 20) wie zum Beispiel Aufgaben (engl. Task), Schleifen (engl. Loop) oder Unterprozesse (engl. Sub-processes).



Abbildung 20: Symbole für verschiedene Arten von Aktivitäten

- **Sequenzfluss**

Ein Sequenzfluss (engl. Sequence Flow) verbindet verschiedene Aktivitäten und zeigt ihre Reihenfolge in einem Prozess (s. Abbildung 21).



Abbildung 21: Das Symbol für einen Sequenzverlauf

- **Ereignis**

Ein Ereignis (engl. Event) hat in der Regel eine Ursache oder Auswirkung und beeinflusst den weiteren Prozessablauf. Ereignisse können am Anfang, am Ende oder mitten in einem Prozess auftreten. Es gibt verschiedene Arten von Ereignissen (s. Abbildung 22), beispielsweise kann ein Ereignis durch das Empfangen einer Nachricht, den Ablauf eines Timers oder das Auftreten eines Fehlers ausgelöst werden. Der Prozess wird dann an der Stelle, an der das Ereignis aufgetreten ist, fortgesetzt.



Abbildung 22: Symbole für die verschiedenen Ereignistypen

- **Gateway**

Ein Gateway steuert den Prozessablauf der Sequenz. Es gibt mehrere Gateway-Typen (s. Abbildung 23):

- Exklusiv: Nur genau eine Untersequenz wird ausgeführt (exklusives oder)
- Inklusiv: Mehrere Untersequenzen können ausgeführt werden (inklusives oder)
- Parallel: Alle Untersequenzen werden parallel ausgeführt.



Abbildung 23: Symbole für verschiedene Arten von Gateways

Beginnend mit Version 2.0 von BPMN wurde die Spezifikation soweit geschärft, dass eine direkte Ausführbarkeit des jeweiligen Prozessmodells ermöglicht wird.

- **Teilnehmer**

Nun fehlt noch die Möglichkeit, den Zuständigen eines (Teil-)Prozesses zu modellieren.



Abbildung 24: Ein Pool mit zwei Lanes

Die Begrifflichkeiten sowie die Darstellung wurden den einzelnen Schwimmbahnen (engl. Swimlane) eines Schwimmbads (engl. Pool) nachempfunden in denen auch je ein Teilnehmer eines Wettbewerbs schwimmt (s. Abbildung 24).

Ein *Pool* stellt eine sogenannte Kooperation dar und kann einen Prozess enthalten. Ein Pool kann auch als Black Box betrachtet werden und keinen Prozess enthalten. Die Teilnehmer sind für den Prozess verantwortlich und für jeden Teilnehmer gibt es eine Swimlane, die den Teilprozess, für den er zuständig ist, enthält.

- **Nachrichtenfluss**

Ein Nachrichtenfluss (engl. Message Flow) zeigt eine Interaktion zwischen einzelnen Pools durch das Senden oder Empfangen von Nachrichten (s. Abbildung 25). Nachrichten werden durch einen Briefumschlag dargestellt. Der Umschlag der ersten (initiiierenden) Nachricht einer Kommunikation ist weiß, während die Antwortnachricht grau eingefärbt wird.

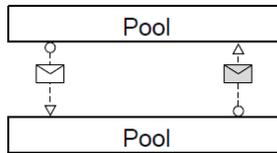


Abbildung 25: Nachrichtenaustausch zwischen Pools

2.4.1.5 Modellierung von Entscheidungsregeln

Mit dem Begriff Entscheidung ist in diesem Zusammenhang ein Test gemeint, der zu einer Entscheidung führt. Dieser erfolgt auf Basis von Entscheidungslogik. Diese Logik wird über eine Reihe von Regeln definiert, die in einer Entscheidungstabelle zusammengeführt sind.

*Definition 26: Einzelne Zeilen einer **Entscheidungstabelle** sind Regeln, und die Entscheidungstabelle ist ein Regelwerk [Oerter, 1968].*

Eine Entscheidungstabelle besteht aus einer Reihe von Entscheidungsregeln, die aus einer Reihe von Bedingungen und einer Reihe von Aktionen bestehen, die ausgeführt werden, wenn die Bedingungen erfüllt sind. In einem vertikalen Layout entspricht jede Spalte einer einzigen Entscheidungsregel [Oerter, 1968]. Entscheidungstabellen können unterschiedliche Formate haben (vgl. Tabelle 7 und 8) und sie können sich in ihrer Interpretation unterscheiden.

Tabelle 7: Entscheidungstabelle mit beschränkten Einträgen

	Entscheidungsregeln				
	1	2	3	4	5
	Bedingungen				
Bedingung 1	Y	Y	Y	N	N
Bedingung 2	Y	Y	N	Y	N
Bedingung ...	Y	N	-	-	-
	Aktionen				
Aktion 1	X		X		X
Aktion 2		X			X
Aktion ...				X	

Tabelle 8: Entscheidungstabelle mit erweiterten Einträgen

	Entscheidungsregeln				
	1	2	3	4	5
	Bedingungen				
Variable 1	<0	<0	>0	>5	-
Variable 2	=1	≠1	<5	<5	-
Bedingung ...	>5	-	-	-	-
	Aktionen				
Prozedur	A	A	B	C	D

Decision Model and Notation (DMN) ist ein relativ neuer Standard⁴¹ zur Modellierung von Regeln bzw. Entscheidungsprozessen. Der Standard stammt wie auch BPMN und UML von der OMG. Die wichtigste Zielsetzung dabei ist ähnlich wie bei BPMN, eine Notation zu definieren, die Fachseite und IT gleichermaßen verstehen [OMG, 2016]. DMN fokussiert bewusst auf die Entscheidungen als Ergebnis und nicht auf die Regeln als notwendige Leitplanken. Die Praxis untermauert dieses Vorgehen: Entscheidungen werden als etwas Positives und Regeln als etwas Negatives wahrgenommen. Deshalb wird nun vermehrt der Begriff *Decision Management* gegenüber

⁴¹ Version 1.0 wurde 2015 veröffentlicht.

dem noch weit verbreiteten Begriff *Business Rules Management* verwendet [Rücker, 2016].

2.4.2 Rahmenwerke für die Systemarchitektur

Ein Rahmenwerk (engl. Framework) stellt Gestaltungsmethoden zur Verfügung und legt für diese Konventionen fest. Solche Konventionen sind beispielsweise Modellierungssprachen, eine Grundstruktur der Architekturelemente und Richtlinien zur Architekturerstellung [Scheckerman, 2004].

*Definition 27: Ein **Architekturrahmenwerk** ist ein Werkzeug, das für die Entwicklung einer breiten Palette unterschiedlichster Architekturen verwendet werden kann. Es sollte ein Verfahren zum Entwurf eines Informationssystems enthalten, welches es ermöglicht, einen Satz von Bausteinen zu beschreiben und zu zeigen, wie die Bausteine zusammenarbeiten. Es sollte ein gemeinsames Vokabular zur Verfügung stellen. Es sollte eine Liste der empfohlenen Standards und kompatibler Produkte umfassen, die verwendet werden können, die Bausteine zu implementieren [TOG, 2011].*

Oft wird auch von Architekturrahmenwerken für Unternehmen (engl. Enterprise Architecture Framework) gesprochen. Der Begriff *Unternehmen* ist jedoch sehr allgemein zu sehen und wäre besser mit *Institution* beschrieben. D.h. diese Art von Rahmenwerken ist geeignet für jede Institution, die das Problem hat, ein komplexes (und meist verteiltes) Informationssystem zu erstellen. Es sind beispielsweise viele Rahmenwerke im Umfeld des Militärs entstanden. Als wesentliche Vertreter klassischer Rahmenwerke sind das Zachman-Framework [Zachman, 1987], TOGAF [OpenGroup, 1995] und das Department of Defense Architecture Framework (DoDAF) [DoD, 2009] zu nennen. Alle drei Rahmenwerke haben bereits eine lange Historie, werden aber bis heute immer wieder

aktualisiert. Stellvertretend wird im folgenden Abschnitt TOGAF vorgestellt, bevor auf ein aktuelleres Rahmenwerk (RM-ODP) eingegangen wird.

2.4.2.1 The Open Group Architecture Framework

Die Architekturarbeitsgruppe des Konsortiums *The Open Group*⁴² entwarf das TOGAF und pflegt es noch heute. Die ursprüngliche Entwicklung der TOGAF Version 1 basierte auf dem *Technical Architecture Framework for Information Management (TAFIM)* des United States Department of Defense [DoD, 1997].

TOGAF unterstützt Entwurf, Planung, Implementierung und Wartung von Unternehmensarchitekturen. Bei der Nutzung von TOGAF wird die Architektur üblicherweise in die drei Domänen Geschäftsarchitektur, Informationssystemarchitektur (unterteilt in Anwendungsarchitektur und Datenarchitektur) und Technologiearchitektur aufgeteilt. Weiterhin definiert TOGAF einen Prozess zur Entwicklung der Architektur, der als *Architecture Development Method (ADM)* bezeichnet wird. Durch die zyklische Ausführung dieses Prozesses wird die Architektur fortgeschrieben.

2.4.2.2 RM-ODP

Das *Reference Model of Open Distributed Processing (RM-ODP)* ist ein Metamodell zur Beschreibung von Informationssystemen, welches in der ISO-Norm ISO/IEC 10746 definiert ist [ISO, 2009]. Das Modell arbeitet mit verschiedenen Sichten (engl. Viewpoints) auf ein System. Die Idee der Modellierung mit Hilfe von Sichten findet sich auch schon bei TOGAF, DoDAF und Zachman. Die von RM-ODP verwendeten Sichten sind in Abbildung 26 dargestellt. Die Erarbeitung der Sichten kann parallel erfolgen, diese müssen jedoch untereinander konsistent sein.

⁴² <http://www.opengroup.org/subjectareas/enterprise/architecture>, letzter Abruf am 4.9.2016

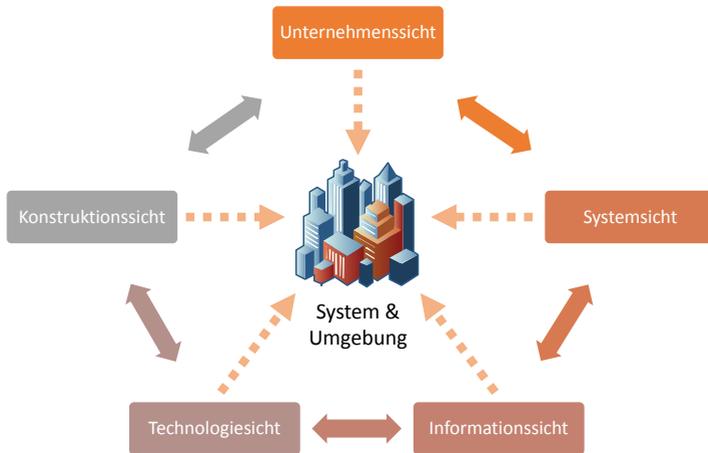


Abbildung 26: Sichten in RM-ODP nach [Dekker, 2008]

In der *Unternehmenssicht* (engl. Enterprise Viewpoint) werden die Anforderungen der Nutzer analysiert und im Hinblick auf ihre Funktions-, Informations- und qualitativen Aspekte dokumentiert. Die *System-sicht* (engl. Computational Viewpoint) beschreibt die im System ablaufenden Prozesse, Berechnungen und Algorithmen. Weiterhin werden die Komponenten des Systems (Dienste, Module, etc.), ihre Schnittstellen und ihr Zusammenwirken dokumentiert. Der Begriff System-sicht ist etwas verwirrend, da diese Sicht ja auch nur einen Aspekt des Systems beleuchtet und wäre treffender mit Komponenten- oder Verarbeitungssicht übersetzt. Da dieser jedoch in allen deutschen Übersetzungen verwendet wird, wird er auch in dieser Arbeit beibehalten. Die *Informationssicht* (engl. Information Viewpoint) klassifiziert und spezifiziert die Informationsanforderungen in Bezug auf ein Informationsmodell. Diese werden technologieunabhängig, zum Beispiel mittels klassischen Entity-Relationship-Diagrammen [Chen, 1976], Ontologien (s. Abschnitt 2.4.1.2) oder UML beschrieben. Die *Technologiesicht* (engl. Technology Viewpoint) legt die Technologien fest, mit denen das System implementiert wird bzw. arbeitet. Insbesondere wird auch

auf verwendete Standards eingegangen. Die Konstruktionssicht (engl. Engineering Viewpoint) bildet die Systemsoftware auf eine konkrete Laufzeitumgebung ab. Zu diskutieren sind u.a. konkrete Hardwareanforderungen, Netzwerktopologien, Installation, etc.

RM-ODP sieht außerdem die Erweiterung durch eigene Sichten vor. In dieser Arbeit wurde beispielsweise die neue Sicht *System von Systemen* eingeführt, um diese spezielle Problematik von Frühwarnsystemen zu adressieren. RM-ODP schreibt keine bestimmten Sprachen oder Notationen für die Beschreibung der einzelnen Sichten vor. Stattdessen ist im Referenzmodell definiert, welche Konzepte verwendet werden sollen, nicht jedoch ihre Darstellung. Dieser notations- und darstellungsneutrale Ansatz wurde bewusst gewählt, um die Einsatzmöglichkeiten und die Flexibilität von RM-ODP zu erhöhen. Allerdings macht diese Flexibilität es schwieriger, Industriewerkzeuge für die Spezifikation, Modellierung und formale Analyse zu entwickeln und die daraus mögliche automatische Ableitung von Implementierungen.

2.5 Vorgehensmodelle

Der Prozess der Softwareerstellung ist eng verbunden mit der Softwarearchitektur. Die für beide Themen existierenden Rahmenwerke bzw. Modelle überlappen sich deshalb häufig. Vorgehensmodelle werden deshalb zur Abgrenzung in diesem Abschnitt kurz vorgestellt, jedoch in den folgenden Kapiteln nicht näher betrachtet. Vorgehensmodelle zur Softwareerstellung fassen Erfahrungen aus dem Management von Softwareprojekten in eine strukturierte Vorgehensweise zusammen. Diese adressieren beispielsweise die immer wieder auftretenden folgenden Fragen:

- Wie strukturiert man ein Projekt zeitlich? Was sind also die Projektphasen und Meilensteine?

- Welche Personen arbeiten im Projekt? Was sind ihre Rollen? Welche Aufgaben haben sie?
- Welche Dokumente und Artefakte sollen zu welchem Meilenstein entstehen?
- Welche Methoden und Werkzeuge werden eingesetzt?

*Definition 28: Ein **Vorgehensmodell** stellt Methoden und Elemente der Softwareentwicklung inklusive des Projektmanagements zu Prozessen und Projektphasen eines standardisierten Projektablaufs zusammen [Hindell, 2004].*

Die verbreitetsten Vorgehensmodelle sind unter anderem das Wasserfallmodell, das Spiralmodell, das V-Modell XT, der Rational Unified Process oder eXtreme Programming. Fritzsche und Keil diskutieren diese in ihrer Arbeit *Kategorisierung etablierter Vorgehensmodelle und ihre Verbreitung in der deutschen Software-Industrie* [Fritzsche, 2007]

3 Stand von Forschung und Technik

In Kapitel 1 wurde gezeigt, dass Frühwarnsysteme eine Unterkategorie verteilter Systeme und Entscheidungssysteme mit einem starken Fokus auf Sensorik sind. Deshalb werden im Folgenden zunächst Architekturen betrachtet, die generell für komplexe verteilte Systeme geeignet sind, danach Architekturen, die als besonderen Fokus die Verwaltung und Integration von Sensoren haben, gefolgt von Architekturen zur Entscheidungsunterstützung und abschließend konkrete Architekturen für Frühwarnsysteme. Nach der Betrachtung von Architekturen wird für den Einsatz semantischer Technologien und die Automatisierung von Arbeitsabläufen in Frühwarnsystemen der Stand der Forschung bei Ontologien und BPMS analysiert.

3.1 Generische Basisarchitekturen

3.1.1 Dienstorientierte Architektur

Die dienstorientierte Architektur (SOA, engl. Service-Oriented Architecture) ist ein Paradigma, das *Dienste* (engl. Services) als fundamentales Element zur Entwicklung verteilter Anwendungen verwendet [Papazoglou, 2004]. Zhang et al. definieren SOA als eine Vorgehensweise für die Architektur von Informationstechnologie, die die Erstellung von Geschäftsprozessen basierend auf funktionalen Einheiten, die als Dienste definiert sind, unterstützt [Zhang, 2007]. Für eine SOA ist also das wichtigste Element der Dienst. Dieser Begriff ist sowohl im Englischen wie auch im Deutschen mit vielfältigen Bedeutungen belegt [Preist, 2004]. Im Rahmen von SOA ist die Definition aus ISO 19119:2016 zutreffend, die frei übersetzt folgendermaßen lautet.

*Definition 29: Ein **Dienst** ist ein Teil der Funktionalität, die von einer Einheit über Schnittstellen zur Verfügung gestellt wird, wobei eine Schnittstelle eine benannte Gruppe von Operationen ist, die das Verhalten der Einheit prägt [ISO, 2016].*

In anderen Worten definiert sich eine SOA im Wesentlichen durch die Schnittstellen ihrer Dienste. Die gleiche Idee wurde bereits bei der Betrachtung von System-of-Systems verwendet, dort allerdings auf Systemebene. Abbildung 27 zeigt die Bereitstellung und Nutzung von Schnittstellen durch Dienste [Gebhart, 2011]. Ein Dienst nutzt zwei andere Dienste mittels ihrer Schnittstelle, um seine Dienstleistung zu implementieren und diese wiederum über eine eigene Schnittstelle zur Verfügung zu stellen.

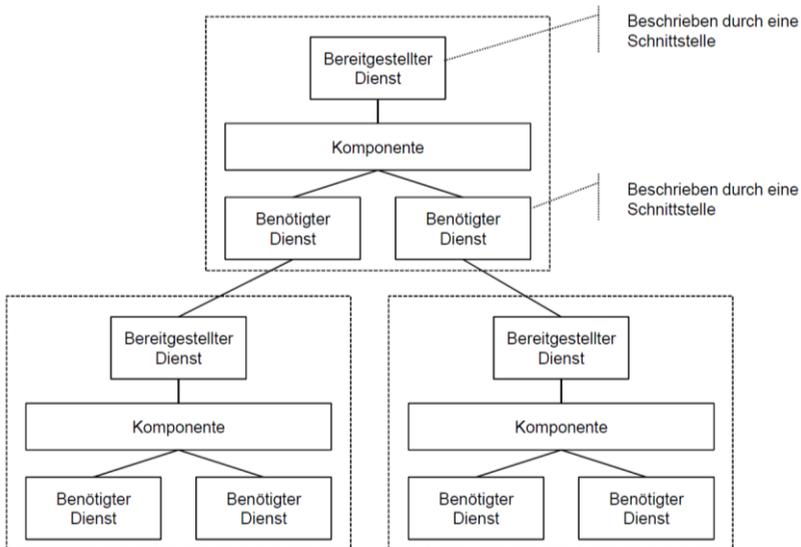


Abbildung 27: Dienste in dienstorientierten Architekturen [Gebhart, 2011]

Die Motivation hinter dienstorientierten Architekturen wurde stark aus dem Unternehmensbereich getrieben, da IT-Abteilungen heutzutage flexibel auf Änderungsanforderungen der betrieblichen Abläufe reagieren müssen [Josuttis, 2008]. Als weitere Vorteile nennt Erl [Erl, 2008] eine höhere Robustheit, Erweiterbarkeit, Flexibilität, Wiederverwendbarkeit und Produktivität als strategische Ziele, die beispielsweise von der Beratungsfirma *Experton Group* bestätigt wurden [Liebhart, 2007]. Tabelle 9 fasst die Vor- und Nachteile von SOA zusammen.

Tabelle 9: Bewertung SOA

Vorteile	Verbesserung der Agilität des Geschäfts und der Produktivität der IT durch die Erhöhung der Anpassungsfähigkeit einer betrieblichen Informatik. Erhöht die Qualitätseigenschaften eines Systems wie Autonomie, Robustheit, Erweiterbarkeit, Flexibilität und Wiederverwendbarkeit beispielsweise durch lose Kopplung (vgl. Abschnitt 2.2.1.2).
Nachteile	Schnittstellenprobleme, wenn nur die Syntax beachtet wird, aber nicht die unterschiedliche Semantik und Ontologie der Begrifflichkeiten und Daten. Verteilte Transaktionen sind eine technische Herausforderung. Bei stark verteilten Diensten kann es schwierig sein, Transaktionssicherheit, Sicherheit, Performance, <i>Quality of Service</i> und <i>Service Level Agreements</i> zu garantieren [Horn, 2016].

3.1.2 Microservices

Microservices stellen einen neuen Trend in der Softwarearchitektur dar, für den es noch keine einheitliche Definition gibt, bzw. größtenteils auch noch kein einheitliches Verständnis vorliegt. Lewis beschreibt Microservices im Interview mit dem IEEE Software Magazin als eine kleine Anwendung, die unabhängig veröffentlicht, skaliert und getestet werden kann und eine einzige Aufgabe erfüllt [Thönes, 2015]. Die einzelne Aufgabe kann nach

seinem Verständnis eine funktionale oder querschnittliche Anforderung an das System sein. Newman teilt diese Auffassung und beschreibt Microservices als kleine, autonome Dienste, die zusammenarbeiten [Newman, 2015].

Die Abgrenzung zu SOA ist nicht leicht. Newman sieht Microservices als eine konkrete Vorgehensweise für SOA, in der die Erfahrungen der letzten Jahre mit SOA berücksichtigt werden [Newman, 2015]. Klassische SOA wird vorwiegend im Backend verwendet, während Microservices auch auf der Ebene der Anwendungsentwicklung liegen [Fowler, 2015]. D.h. ein Microservice enthält damit auch eine eigene Benutzerschnittstelle, die sich jedoch in die Benutzeroberfläche der Gesamtanwendung integrieren muss [Thönes, 2015].

Tabelle 10: Bewertung Microservices

Vorteile	Für Microservices werden die gleichen Vorteile angeführt wie bei der klassischen SOA (s. Abschnitt 3.1.1). Oft wird auf den Vorteil der klaren Größe und Funktionalität eines Microservice hingewiesen. Dadurch soll auch die Autonomie der unterschiedlichen Entwicklerteams gefördert werden [Killalea, 2016].
Nachteile	Microservices werden selbst von ihren Befürwortern kritisch gesehen und stellen keine Lösung für jede Anwendung dar [Fowler, 2015]. Insbesondere werden Entwickler nun auf der Anwendungsebene mit sämtlichen Problemen von verteilten Systemen konfrontiert wie beispielsweise der verteilten Datenhaltung, verteilter Autorisierung, asynchroner Kommunikation und instabilen Netzen [Newman, 2015]. Die Umstellung des Quellcodes (engl. Refactoring) ist viel schwieriger als mit normalen Bibliotheken, da Code über Servicegrenzen hinwegbewegt werden muss. Alle Änderungen an der Benutzungsoberfläche müssen zwischen den Entwicklerteams abgestimmt werden [Fowler, 2015].

3.1.3 Ereignisgesteuerte Architektur

Eine ereignisgesteuerte Architektur (EDA, engl. Event-driven Architecture) ist eine Softwarearchitektur, in der das Zusammenspiel der Komponenten durch Ereignisse gesteuert wird. Michelson definiert ein Ereignis wie folgt [Michelson, 2006].

*Definition 30: Ein **Ereignis** ist eine wichtige Sache, die innerhalb oder außerhalb eines Unternehmens geschieht. Ein (Geschäfts- oder System-) Ereignis kann ein Problem (oder ein drohendes Problem), eine Möglichkeit, ein Schwellenwert oder eine Abweichung bedeuten.*

Bruns und Dunkel setzen EDA in enge Beziehung mit Complex Event Processing (CEP): EDA und CEP repräsentieren einen neuen Stil von Unternehmensanwendungen, bei dem Ereignisse in das Zentrum der Softwarearchitektur rücken. Während EDA einen ereignisorientierten Entwurstil für Anwendungsarchitekturen beschreibt, handelt es sich bei CEP um eine Technologie zur dynamischen Verarbeitung von großen Volumina von Ereignissen in Echtzeit [Bruns, 2010]. Aus technischer Sicht stellt die EDA ein flexibles Verarbeitungsmodell innerhalb einer entkoppelten Systemstruktur dar. Ihre Komponenten kommunizieren asynchron über *Publish & Subscribe* [Luckham, 2012], d.h.:

- Ereignisquellen erkennen relevante Informationen, erstellen daraus eine formale Darstellung und veröffentlichen diese an einen Mediator, ohne Kenntnis der weiteren Verarbeitungsschritte.
- Der Mediator verwaltet die Ereignisobjekte in einem Ereigniskanal. Er organisiert die Verteilung der Ereignisobjekte an die Ereignissenken.
- Ereignissenken empfangen Ereignisobjekte und verarbeiten diese mittels einer vordefinierten Geschäftslogik, die nur ihnen bekannt ist. Eine solche Komponente kann wiederum selbst neue Ereignisse veröffentlichen und somit als Ereignisquelle auftreten.

Message-oriented Middleware

Eine Message-oriented Middleware (MOM) bietet eine Schicht zwischen der Transport- und Anwendungsschicht des Netzwerk-Protokoll-Stacks zum Austausch von Meldungen [Wang, 2010]. Sie ermöglicht Anwendungen und verteilten Systemen in heterogenen Umgebungen durch den Austausch von Nachrichten zu kommunizieren. Die MOM-Infrastruktur unterstützt nicht nur das synchrone Kommunikationsmodell, sondern ermöglicht auch einen asynchronen Informationsaustausch, der in vielen Fällen einer zeitlich und räumlich verteilten Anwendungsintegration und Informationsverbreitung vorzuziehen ist [Pietzuch, 2003]. Darüber hinaus ermöglicht eine MOM Anwendungen oder Systemen Nachrichten mit anderen Anwendungen oder Systemen auszutauschen, ohne Details über Plattformen und Netze der Gegenseite wissen zu müssen, was die Interoperabilität, Portabilität und Flexibilität erhöht. Eine EDA kann mit Hilfe einer MOM umgesetzt werden [Moßgraber, 2013]. Tabelle 11 fasst die Vor- und Nachteile von EDA zusammen.

Tabelle 11: Bewertung EDA

Vorteile	<p>EDA ermöglicht eine hohe Flexibilität, wenn Aktionen auf Echtzeitereignisse erfolgen müssen. Krumeich sieht deshalb EDA als Voraussetzung für das <i>echtzeitfähige Unternehmen</i> an [Krumeich, 2014].</p> <p>Ein großer Vorteil einer EDA ist laut Luckham die Tatsache, dass dieses Konzept keine ausschließenden Gegensätze zu anderen Architekturstilen aufweist und folglich sehr gut als ergänzende Komponente zu implementieren ist [Luckham, 2012] (vgl. Event-driven SOA).</p>
Nachteile	<p>Als Nachteil führt Bruns auf, dass sich das strukturierte Testen von EDA-Anwendungen sehr schwierig gestaltet, da der Kontrollfluss undurchsichtig ist [Bruns, 2010].</p>

3.1.4 Event-Driven SOA

Bei der Beschreibung von EDA wurde schon auf die leichte Kombinierbarkeit mit anderen Architekturkonzepten hingewiesen, insbesondere mit SOA. Dieser kombinierte Einsatz wird in der Literatur als *Event-Driven SOA* oder auch als *SOA 2.0* bezeichnet [Chou, 2008], [Bruns, 2010]. Mit der Kombination beider Architekturen sollen die jeweiligen Vorteile ebenfalls kombiniert werden [Luckham, 2012]. Das Ziel einer komplementären Lösung ist die überbetriebliche Verbindung mehrerer heterogener IT-Systeme (s. Abbildung 28).

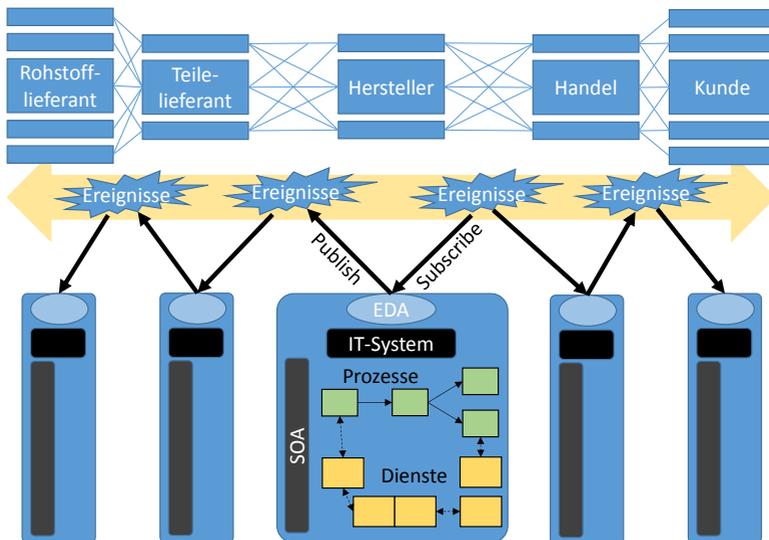


Abbildung 28: Kombination der Konzepte von EDA und SOA nach [Buckel, 2012]

Für die unternehmensinterne Flexibilisierung der Systemlandschaft und Gestaltung der Prozesse ist eine SOA von Vorteil (untere Ebene). Löst ein bestimmter Dienst aufgrund seiner Zustandsänderung ein Ereignis aus (beispielsweise fehlende Teile), sind basierend auf einer orthogonalen EDA-

Erweiterung neue SOA-Domänen weiterer IT-Systeme zu erreichen (Teile werden bestellt). Der Dienst ist somit ebenfalls ein ereignisproduzierender Baustein der überbetrieblichen EDA. Durch diese ereignisbasierte Kopplung der verschiedenen, orchestrierten Dienste der SOA erreicht die gesamte Abwicklung innerhalb des Netzwerks eine hohe Flexibilität und ermöglicht die Anpassungsfähigkeit gegenüber auftretenden Ereignissen [Buckel, 2012]. Die Bewertung der Event-Driven SOA Architektur ergibt sich aus den Vor- und Nachteilen von EDA und SOA und wird an dieser Stelle nicht wiederholt.

3.1.5 Lambda Architektur

Die *Lambda Architektur* ist eine datenverarbeitende Architektur, die entworfen wurde, um Massendaten zu handhaben. Sie verbindet die Vorteile von Stapel- und Streaming-Verarbeitungsverfahren und versucht somit eine Balance zwischen Latenz, Durchsatz und Fehlertoleranz zu finden (s. CAP-Theorem in Abschnitt 2.2.3.1).

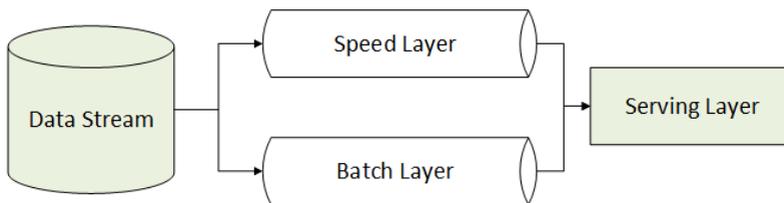


Abbildung 29: Lambda Architektur nach Marz [Marz, 2015]

Die Architektur ist in Abbildung 29 dargestellt. Die langsamere Stapelverarbeitung nimmt eine tiefgehende Analyse der eingehenden Daten vor (*Batch Layer*), während gleichzeitig in Echtzeit eine gröbere Echtzeitverarbeitung stattfindet (*Speed Layer*), die einen schnellen Überblick zur Verfügung stellt. Die Ergebnisse der beiden Sichten können vor der Darstellung kombiniert werden (*Serving Layer*) [Marz, 2015].

Tabelle 12 fasst die Vor- und Nachteile der Lambda Architektur zusammen.

Tabelle 12: Bewertung Lambda Architektur

Vorteile	Ermöglicht die Anpassung von Geschwindigkeits- und Tiefenanalyse in der Datenverarbeitung an die Bedürfnisse des eigenen Systems (Ausbalancierung des CAP-Theorems).
Nachteile	Kreps kritisiert, dass für das Speed und Batch Layer unterschiedliche Implementierungen und Werkzeuge benötigt werden. Bereits eines der Layer besitzt die hohe Komplexität von verteilten Systemen und die Lambda Architektur führt nun zur Doppelarbeit und -pflege, die diese Komplexität vervielfacht [Kreps, 2014].

3.2 Architekturen mit Schwerpunkt Sensorik

Alle der folgenden Architekturen arbeiten mit einem erweiterten Sensorbegriff. Dieser beinhaltet neben physikalischen Sensoren auch sogenannte virtuelle Sensoren (engl. Virtual Sensors).

*Definition 31: Ein **virtueller Sensor** ist ein Software-Sensor (im Gegensatz zu einem physikalischen oder Hardware-Sensor). Virtuelle Sensoren liefern indirekte Messungen abstrakter Bedingungen, die aus den Messwerten einer Menge heterogener physikalischer Sensoren fusioniert bzw. berechnet werden [Kabadayi, 2006].*

3.2.1 Internet of Things

Computerisierte Mess- und Regeltechnik sowie Mechatronik und Robotik erweitern nicht nur bestehende Systeme, sondern greifen tiefgehend in Abläufe ein. Aus konventionellen Systemen entstehen intelligente und kooperierende Systeme. Durch Kommunikation über Internettechnologien

können Menschen mit Maschinen, Maschinen mit Clouddiensten, oder Maschinen mit Maschinen interagieren. Das Internet der Dinge (IoT, engl. Internet of Things) bildet dabei den Sammelbegriff in Hinblick auf Konnektivität, Datenerfassung und BigData. Da IoT noch sehr neu ist, gibt es aktuell viele konkurrierende Ansätze, Technologien und in Entstehung begriffene Standards. Das EU-Projekt *symbloTe* (symbiosis of smart objects across IoT environments)¹ untersucht aktuell, wie die gängigsten IoT-Plattformen aufeinander abgebildet werden können. Im Folgenden werden stellvertretend für die vielfältigen Ansätze zwei Architekturen für IoT vorgestellt, die auf Grund ihrer Konsortien Aussicht auf Erfolg haben könnten.

3.2.1.1 IoT-A

Das EU-Projekt *Internet of Things – Architecture (IoT-A)*² definierte ein Referenzmodell für IoT Architekturen. Auf abstrakter Ebene wurden die wesentlichen Elemente, die eine IoT Architektur benötigt, identifiziert und Empfehlungen gegeben, wie diese in eine konkrete Systemarchitektur umgesetzt werden kann [Carrez, 2013]. Bei IoT-A handelt es sich um eine SOA und nutzt das OASIS Referenzmodell für dienstorientierte Architekturen als Basis [MacKenzie, 2006]. Dies spiegelt sich in der Systemsicht wieder, die in Abbildung 30 dargestellt ist.

¹ <https://www.symbiote-h2020.eu/>, letzter Abruf am 4.9.2016

² <http://www.iot-a.eu/>, letzter Abruf am 4.9.2016

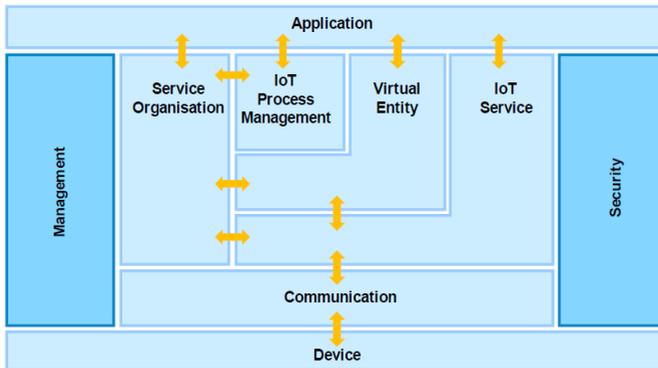


Abbildung 30: Systemsicht der IoT Reference Architecture (IoT-A) [Carrez, 2013]

Zentraler Bestandteil ist die *Service Organisation* zur Komposition, Orchestrierung, Choreographie und dem Auffinden von Diensten sowie die Definition der Dienste (*IoT Service*) selbst.

Als Basis der IoT-Dienste dienen mehrere Kommunikationsarten mit den Geräten (*Devices*). Geräte können mehrere Sensoren und auch Aktoren zur Steuerung enthalten. Ein virtueller Sensor wird als *Virtual Entity* bezeichnet. Verwaltungs- und Sicherheitsfunktionalität sind als Querschnittsdienste vorgesehen [Carrez, 2013]. Tabelle 13 fasst die Vor- und Nachteile von IoT-A zusammen.

Tabelle 13: Bewertung von IoT-A

Vorteile	IoT-A enthält die wesentlichen Komponenten für eine IoT Architektur.
Nachteile	Da es sich um ein Rahmenwerk für Architekturen handelt, muss IoT-A in der Anwendung erst noch in eine konkrete Architektur überführt werden. Hierbei erhält der Systemarchitekt jedoch wenig Hilfestellung, welche konkreten Technologien verwendet werden können. Das Informationsmodell basiert auf keinem Standard und muss im konkreten Fall für die zu verwendenden Geräte ausgearbeitet werden. Die Interaktion der Querschnittsdienste Sicherheit und Management mit den IoT Diensten bleibt unklar.

3.2.1.2 Industrial Internet of Things

Das *Industrial Internet Consortium*³ (IIC) ist ein Zusammenschluss namhafter Firmen wie zum Beispiel AT&T, Cisco, GE, IBM und Intel zur Erstellung eines Architekturrahmenwerks für industrielle IoT Anwendungen.

Die aktuelle Spezifikation in Version 1.7 bewegt sich aktuell noch auf der Granularität von Abbildung 31 [IIC, 2015]. Deshalb ist zum jetzigen Zeitpunkt noch keine Bewertung dieser Architektur möglich.

³ <http://www.iiconsortium.org/>, letzter Abruf am 4.9.2016

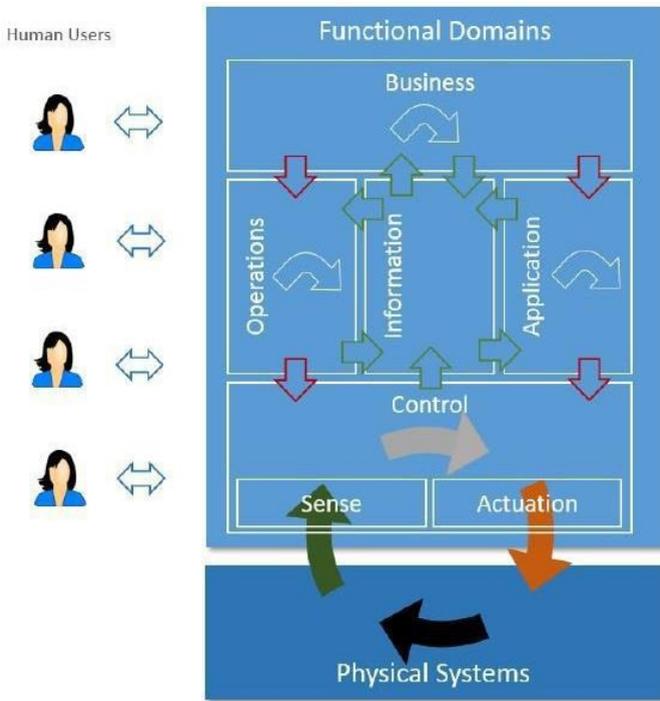


Abbildung 31: Funktionale Sicht der IIC Architektur [IIC, 2015]

Die Hauptelemente der Architektur besitzen einen starken Fokus auf industrielle Steuersysteme. Diese Elemente werden in der Spezifikation als *Domänen* bezeichnet. Die Steuerungsdomäne (engl. Control Domain) stellt Funktionen für industrielle Steuerungssysteme zur Verfügung. Die Operations Domain dient zur Bereitstellung, Verwaltung, Überwachung und Optimierung der Systeme in der Steuerdomäne. Die Informationsdomäne (engl. Information Domain) sammelt Daten aus verschiedenen Bereichen (insbesondere der Steuerdomäne) und transformiert, persistiert und analysiert diese, um auf hoher Ebene Auswertungen über das Gesamtsystem zu erhalten. Die Anwendungsdomäne (engl. Application Domain) implementiert schließlich die Geschäftsprozesse, die auf dem

Informationsmodell aufbauen und verbindet diese in der Businessdomäne mit existierenden Informationssystemen des Unternehmens.

3.2.2 Sensor Web Enablement

Die SWE Initiative des OGC mit ihren vielzähligen Standards wurde bereits vorgestellt. Es wurde jedoch noch nicht gezeigt, wie die Interaktion der SWE- Dienste und Kodierungsstandards vorgesehen ist. Abbildung 32 und Abbildung 33 zeigten diese Interaktion schematisch und Abbildung 34 für ein konkretes Anwendungsszenario.

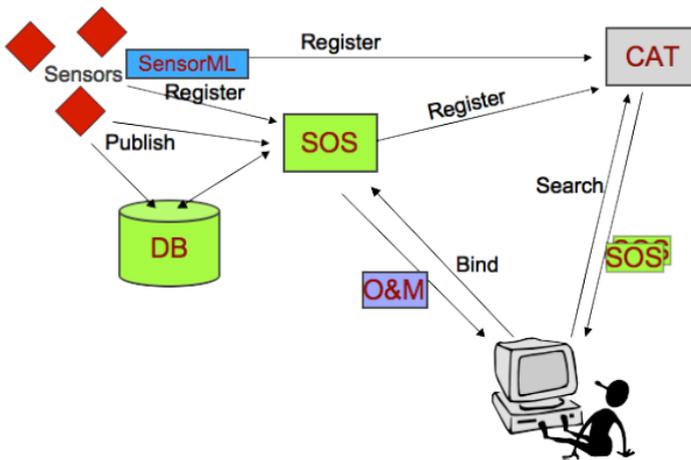


Abbildung 32: Interaktion der SWE Dienste und Kodierungen - Teil 1 [Simonis, 2008]

In der oberen linken Ecke von Abbildung 32 sieht man Sensoren, die in einem SOS registriert sind. Veröffentlichen diese ihre Beobachtungsergebnisse zu den Diensten selbst, handelt es sich um einen *Transactional-SOS*, andernfalls werden diese in eine Datenbank geschrieben, auf die von der SOS-Dienstinstanz zugegriffen wird. Um Sensoren auffindbar zu machen,

werden diese mit SensorML beschrieben und inkl. ihrem SOS bei einem Katalogdienst (CAT) registriert. Der Benutzer in der unteren rechten Ecke möchte die Beobachtungsdaten nutzen und sendet daher eine Anfrage zum Katalog. Der Katalog antwortet mit einer Liste von SOS-Dienstinstanzen, die die gewünschten Anforderungen erfüllen. Schließlich verbindet sich der Benutzer mit dem SOS und ruft die Beobachtungsdaten ab, welche in O&M codiert sind.

Abbildung 33 zeigt ein komplexeres Szenario. In diesem werden die Sensoren mittels eines SPS konfiguriert. In diesem Szenario wird beispielsweise eine Kamera ausgerichtet. WNS wird verwendet, um über neue Daten, beendete Konfigurationen oder Berechnungen zu informieren.

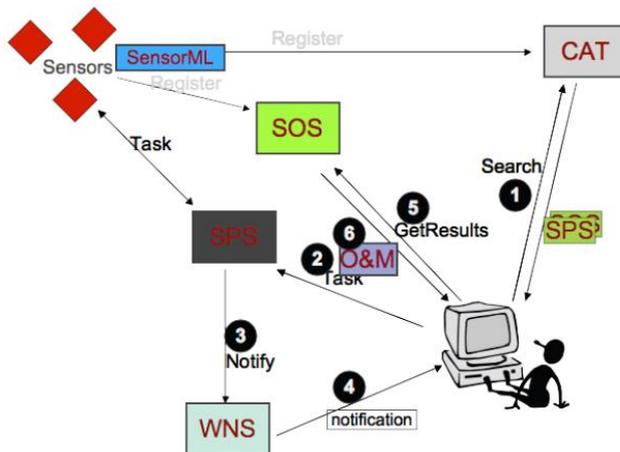


Abbildung 33: Interaktion der SWE Dienste und Kodierungen - Teil 2 [Simonis, 2008]

Beim Einsatzbeispiel in Abbildung 34 kommt als neues Element noch ein SES hinzu. Sind für einen Anwendungsfall nur Daten, die den definierten Filterkriterien entsprechen, relevant, beispielsweise die Überschreitung von Grenzwerten, kann hierzu ein SES abonniert werden. In Kombination mit

einem WNS können Alarme versandt werden, falls bestimmte Ereignisse auftreten. Zum Beispiel kann ein Benutzer eine Benachrichtigung per SMS oder E-Mail erhalten, wenn der Wasserstand bei einer Messstation über fünf Meter steigt. Tabelle 14 fasst die Vor- und Nachteile von SWE zusammen.

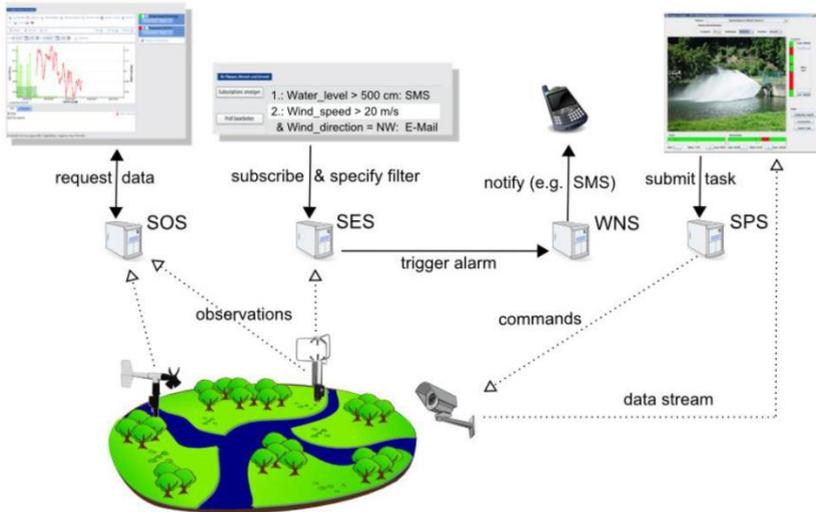


Abbildung 34: Einsatzbeispiel für SWE Dienste [Bröring, 2011]

Tabelle 14: Bewertung SWE

Vorteile	SWE löst vielfältige Probleme bei der Nutzung von Sensorik und ihrem Management.
Nachteile	Die SWE Dienste haben sich insbesondere durch ihre umfangreiche Nutzung von XML-Strukturen als schwerfällig erwiesen und konnten sich deshalb in der Praxis bisher nicht durchsetzen. Das OGC strebt mittlerweile auch leichtgewichtige Lösungen an, wie beispielsweise die SensorThings API.

3.2.3 SANY

Das EU-Projekt SANY konzentrierte sich auf die Interoperabilität von in-situ-Sensoren und Sensornetzen und die leichte Verarbeitbarkeit der Sensordaten, sodass sie als Grundlage für die Entscheidungsunterstützung verwendet werden können. Als architektonischen Rahmen definierte das SANY Projekt die Sensor-Service-Architektur (*SensorSA*), die zur Entwicklung sensorbasierter Umwelthanwendungen dient und in Abbildung 35 dargestellt ist.

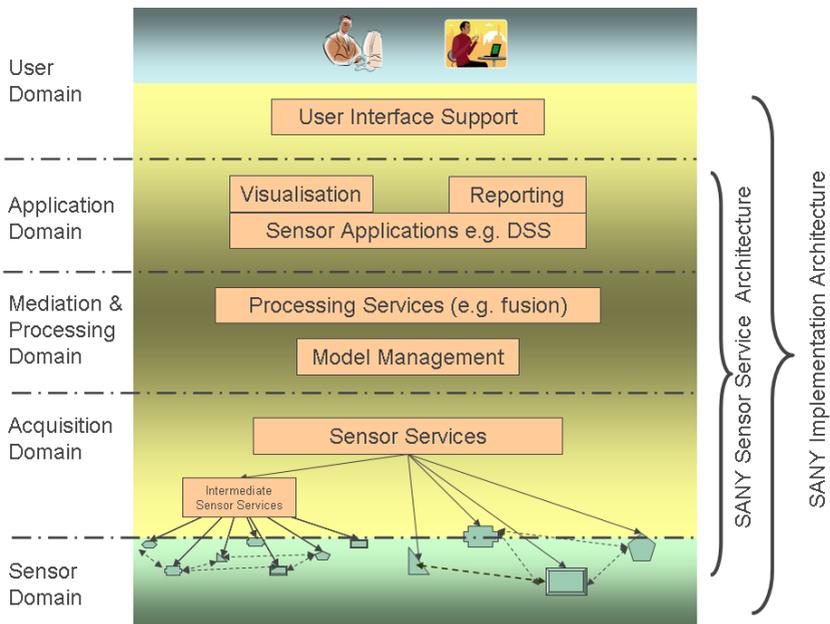


Abbildung 35: Die SANY Schichtenarchitektur [Usländer, 2009]

Die SensorSA ist eine serviceorientierte Architektur mit einem besonderen Fokus auf den Zugriff, die Verwaltung und Verarbeitung von Informationen von Sensoren und Sensornetzen [Usländer, 2009]. Die Grundlage für die

SensorSA ist die ORCHESTRA Projekt Architektur (RM-OA) [Usländer, 2007] und die OGC SWE Architektur. Die Architektur ist als klassische Schichtenarchitektur gehalten bestehend aus fünf Schichten. Auf der untersten Ebene befinden sich die Hardware-Sensoren, darüber der Zugriff auf Sensoren und ihre Datenabfrage, gefolgt von Verarbeitungsdiensten und schließlich domänenspezifischen Anwendungsdiensten mit Benutzerschnittstellen.

Alle Sensormessdatensätze (Rohdaten) in SANY wurden jeweils von einem SOS bereitgestellt. Weiterhin wurden auch die Ergebnisse der Fusionsverarbeitung durch SOS Schnittstellen verfügbar gemacht. Beispiele für Sensordatensätze, die in SANY verwendet wurden, sind Messungen für Luftverschmutzung, Meteorologie und Bodenverschiebungen. Wenn ein WPS oder ein SPS Daten benötigt, verwendet er einen SOS-Client. Abbildung 36 zeigt welche OGC-Dienstschnittstelle zum Zugriff eines Verarbeitungsschritts genutzt werden kann.

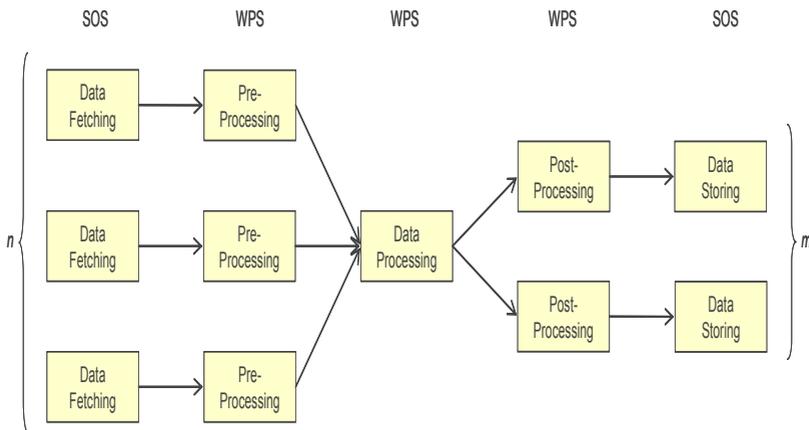


Abbildung 36: Beispiel für SANY Verarbeitungskette mittels OGC-Diensten [Middleton, 2010]

Für die Datenmodellierung wurde das OGC O&M Modell mit der Uncertainty Markup Language (UncertML⁴) verbunden, um neben den reinen Sensordaten auch Informationen über Einheiten, Messarten sowie die Sensorgenauigkeit zur Verfügung stellen zu können. Ein großer Teil der Vorverarbeitung für die Datenfusion konnte mittels dieser Informationen automatisiert werden. Beispiele hierfür sind die Zusammenführung der Datensätze identischer Phänomene aus mehreren SOS sowie die Validierung der Wertebereiche der Daten durch die Kenntnis der jeweiligen Einheit. Zur Steuerung mehrstufiger Verarbeitungsketten, die verschiedene SOS, WPS und SPS nutzen, wurde BPEL eingesetzt [Middleton, 2010].

In SANY wurden keine BigData-Konzepte integriert, jedoch wurde der Transport großer Mengen von Messwertdaten als Problem erkannt. Da SOS den Zugriff und das Format der eigentlichen Messdaten offenlässt, wurde als Lösung die binäre Speicherung mittels *Network Common Data Form* (NetCDF) [Nativi, 2015] und als Transportprotokoll das *Open-source Project for a Network Data Access Protocol* (OPeNDAP) [Cornillon, 2003] vorgeschlagen. NetCDF umfasst ein Datenmodell für wissenschaftliche Messreihen, ein maschinenunabhängiges Format sowie eine Sammlung von Softwarebibliotheken zum Zugriff auf das Datenmodell. Tabelle 15 fasst die Vor- und Nachteile der SANY Architektur zusammen.

⁴ <http://www.uncertml.org>, letzter Abruf am 4.9.2016

Tabelle 15: Bewertung SANY Architektur

Vorteile	SANY erweitert die Lösungen der SWE-Architektur um ein Konzept zur generischen Integration von Verarbeitungsabläufen mittels BPEL. Weiterhin werden Lösungsmöglichkeiten für DSS Anwendungen und Benutzerschnittstellen aufgezeigt, die jedoch nicht Teil der generischen SANY-Architektur sind, sondern in speziellen Anwendungsfällen experimentell erprobt wurden.
Nachteile	Als Erweiterung der SWE-Architektur bringt die SANY-Architektur die gleichen Nachteile (s.o.) mit sich. Die interessante Idee, ein BPMS zur Orchestrierung der Verarbeitungsdienste zu nutzen, erwies sich durch die Wahl von BPEL als komplex und schwerfällig.

3.3 Architekturen zur Entscheidungsunterstützung

Im Folgenden werden Architekturen für Entscheidungsunterstützungssysteme und die spezielleren Expertensysteme betrachtet. Da es sich um sehr grobe Architekturen handelt, erfolgt keine Bewertung.

3.3.1 Architektur für Entscheidungsunterstützungssysteme

In der Literatur findet man unterschiedliche Meinungen, welche Komponenten ein Entscheidungsunterstützungssystem benötigt [Bröring, 2011], [Pinto, 2014].

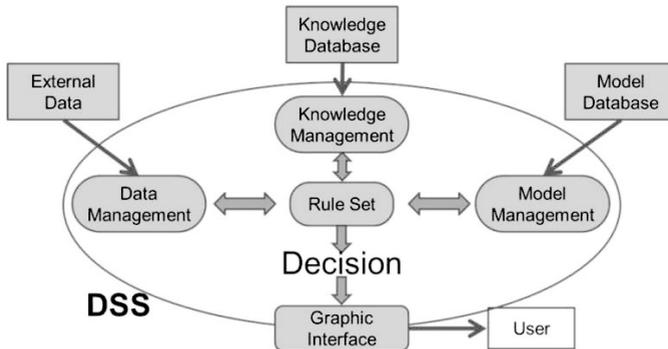


Abbildung 37: Komponenten eines Entscheidungsunterstützungssystems [Pinto, 2014]

Abbildung 37 zeigt die DSS Architektur nach Pinto, die folgende Komponenten enthält: *Externe Daten* (engl. External Data) werden beispielsweise von Sensoren oder Informationssystemen zugeliefert. Sie geben Auskunft über den aktuellen oder früheren Status von Variablen oder Phänomenen, die von Interesse sind. *Modelle* (engl. Model) werden verwendet, um das Verhalten von Phänomenen ausgehend von den aktuell über sie verfügbaren Informationen zu prognostizieren. Das hinterlegte *Wissen* (engl. Knowledge) stammt aus Erfahrungen oder Informationen von Experten und ermöglicht das Verständnis der Beziehungen, die die beobachteten oder modellierten Daten haben können. Konsens herrscht darin, dass ein DSS eine Benutzerschnittstelle benötigt, Modelle notwendig sind und das Wissen verwaltet werden muss.

Der *Regelsatz* (engl. Rule Set) ermöglicht die Kombination der Informationen, um eine Entscheidung über die Fragestellungen zu erhalten. Die *Benutzerschnittstelle* (engl. User Interface) erlaubt dem Benutzer die Interaktion mit dem DSS, um die aktuelle Situation zu verstehen und Ratschläge vom System zu erhalten. Im Handbuch für DSS [Bröring, 2011] findet man eine verfeinerte Darstellung, die in Abbildung 38 dargestellt ist.

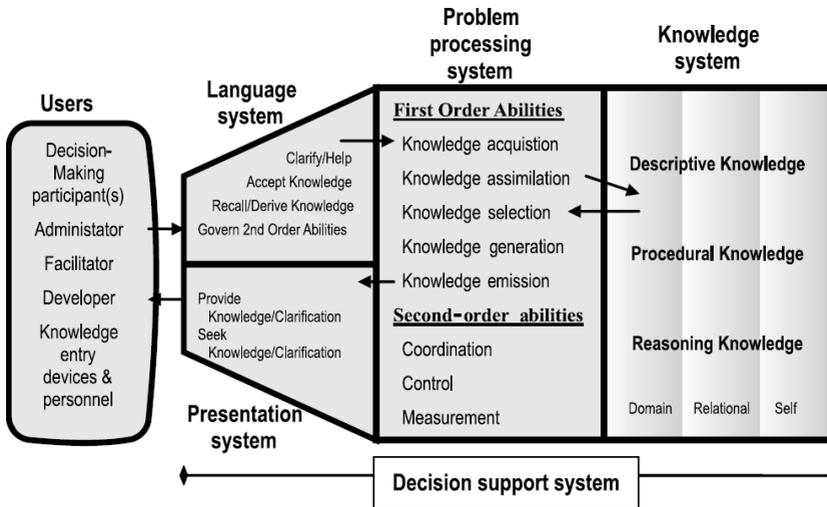


Abbildung 38: Komponenten eines Entscheidungsunterstützungssystems [Bröring, 2011]

Das Wissen wird in drei Bereiche geteilt. Es kann in beschriebener (Faktenwissen), prozeduraler (Funktionen) oder abgeleiteter (durch Regeln aus dem Faktenwissen) Form vorliegen. Die Benutzerschnittstelle wird untergliedert in die Visualisierungsunterstützung zum Verständnis eines Problems oder einer Situation (engl. Presentation System) und das Sprachsystem (engl. Language System) zur Interaktion mit dem DSS. Den Kern stellt die Komponente zur Erkennung und Bearbeitung von Problemen (engl. Problem Processing System) dar.

3.3.2 Architektur für Expertensysteme

Wie in Abschnitt 2.2.3.2 beschrieben wurde, sind Expertensysteme in der Regel eigenständige Problemlösungssysteme für Situationen, in denen das Problem klar definiert ist und das Expertenwissen angewendet werden kann, um die passende Lösung zu finden. Dies spiegelt sich wider in der in Abbildung 39 gezeigten Architektur [Bröring, 2011]. Experten bauen eine

Wissensbasis (engl. Knowledge Base) ihres Fachwissens auf. Aus diesem Wissen kann eine Inferenzmaschine bestimmte Fragestellungen beantworten. Zur Umsetzung werden meist Entscheidungsbäume eingesetzt.

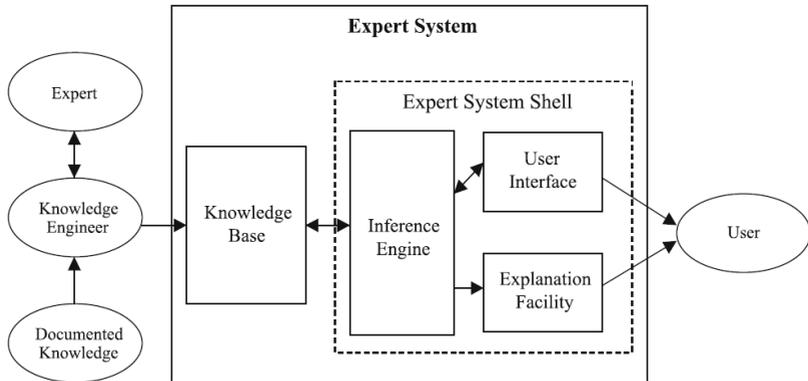


Abbildung 39: Architektur für Expertensysteme nach [Bröring, 2011]

3.4 Architekturen für Frühwarnsysteme

Die folgenden Architekturen werden untergliedert in generische Aspekte für Frühwarnsystemarchitekturen und spezifische Architekturen für Frühwarnsysteme vor Erdbeben, Tsunami und Flut.

3.4.1 Generische Architekturen

Das Hauptziel von Frühwarnsystemen ist eine präzise Bereitstellung von Informationen im Falle einer gefährlichen Situation, um reagieren zu können bevor Schäden entstehen, bzw. um die Höhe der Schäden zu minimieren. Hierzu müssen EWS verschiedene Aufgaben erfüllen [Turoff, 2002], [Lindel, 2007]. Diese Funktionalitäten gliedern Meissen und Voisard in eine grobe Schichtenarchitektur, die in Abbildung 40 dargestellt ist [Meissen, 2008].

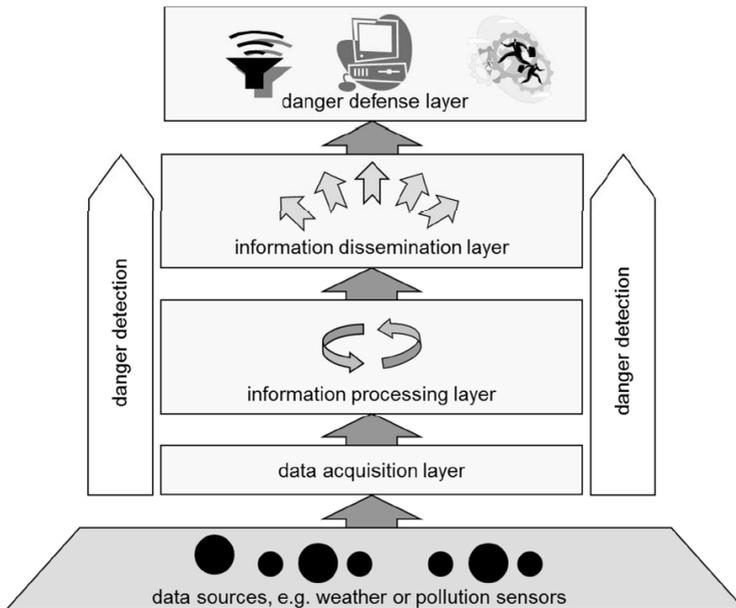


Abbildung 40: EWS als Schichtenarchitektur [Meissen, 2013]

Diese umfasst folgende Schichten:

Datenerfassungsschicht (engl. Data Acquisition Layer): Sensoren oder Sensor-Netzwerke sammeln Daten über Umweltveränderungen, um Risiken oder Gefahren zu erkennen. Zusätzlich wird eine Aggregation von mehreren physikalischen Messungen durchgeführt. Eine Vorfilterung irrelevanter Informationen in der Datenerfassungsschicht verringert die Kommunikationslast des Systems und verbessert die Leistung in den anderen Schichten.

Informationsverarbeitungsschicht (engl. Information Processing Layer): In dieser Schicht werden die gesammelten Daten analysiert, um Prognosen zur Wahrscheinlichkeit, Zeit, Dauer und Auswirkungen einer Gefahr und möglichen Schäden zu erstellen.

Informationsverbreitungsschicht (engl. Information Dissemination Layer): Diese Schicht wandelt die Informationen über eine Gefahr in eine Warnmeldung um. Diese Meldung ist jeweils sprachlich an die verschiedenen Empfänger angepasst und wird über die verfügbaren Kommunikationsmittel verteilt.

Schicht zur Gefahrenabwehr (engl. Danger Defense Layer): Diese Schicht stellt geeignete Werkzeuge bereit, um Notfallmaßnahmen zu planen, zu initiieren, durchzuführen und zu steuern. Nutzer können Menschen oder auch IT-Systeme der technischen Anlagen sein. IT-Systeme können die Verbindung und Interaktion zwischen den verschiedenen Beteiligten durch den Austausch von Informationen unterstützen.

3.4.1.1 Reference Architecture for Early Warning Systems

Meissen und Voisard abstrahieren die von ihnen in den Projekten German Indonesian Tsunami Early Warning System (GITEWS⁵) und Weather Information on Demand (WIND) [Meissen, 2013] gewonnenen Erfahrungen bei der Architektur von Frühwarnsystemen in ihrer Veröffentlichung *Towards a Reference Architecture for Early Warning Systems* zu einer generischen Architektur [Meissen, 2006]. Wie aus dem Titel schon erkennbar ist, handelt es sich nicht um eine vollständig ausgearbeitete Architektur für EWS, sondern vertieft bestimmte Aspekte. Angelehnt an die Phasen der Frühwarnung (s. Abschnitt 2.1.4) wird definiert, dass ein EWS aus Subsystemen besteht, die in 4 Kategorien eingeteilt werden, welche in sequentieller Folge kommunizieren (s. Abbildung 41):

⁵ S. Abschnitt 3.4.3.1

1. Überwachungssysteme (engl. Monitoring systems)
2. Gefahrenerkennungssysteme (engl. Hazard detection systems)
3. Risikoanalysensysteme (engl. Risk assessment systems)
4. Warnsysteme (engl. Warning Systems)

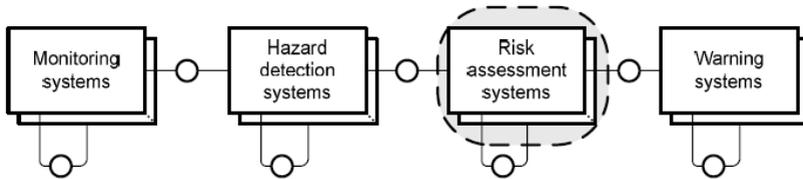


Abbildung 41: Subsysteme eines Ende-zu-Ende EWS [Meissen, 2006]

Eine Architektur für die Kategorie Risikoanalysensysteme wird von den Autoren nicht betrachtet und deshalb in der Abbildung gestrichelt umrahmt. Die Architekturen für Kategorie 1 und 2 stammen aus GITEWS und für Kategorie 4 aus WIND. Da die Autoren der Architektur keinen Namen geben, wird diese im Folgenden als RA-EWS (Reference Architecture for Early Warning Systems) bezeichnet. Tabelle 16 fasst die Vor- und Nachteile der RA-EWS Architektur zusammen.

Tabelle 16: Bewertung RA-EWS

Vorteile	RA-EWS stellt einen ersten Ansatz einer Architektur für ein vollständiges Ende-zu-Ende Frühwarnsystem dar.
Nachteile	<p>RA-EWS ist nicht vollständig:</p> <p>Die Architektur für die Kategorie Risikoanalysesysteme wird von den Autoren nicht betrachtet.</p> <p>Es wird nur die Systemsicht betrachtet (im Sinne von RM-ODP), d.h. weder das Informationsmodell noch die Verteilung werden thematisiert.</p> <p>Die Kommunikation und der Datenaustausch der Untersysteme werden nicht erläutert. Weiterhin ist unklar, wie beispielsweise eine Integration von Benutzeroberflächen stattfinden kann bzw. soll.</p> <p>Die sequentielle Kommunikation der Untersysteme, angelehnt an die Phasen der Frühwarnung, entspricht nicht mehr dem Stand der Forschung, die eine Verzahnung und Parallelität fordert (vgl. 2.1.1.1). Die Funktionalitäten der Gefahrenabwehr (vgl. Abbildung 40) finden keine Berücksichtigung.</p>

3.4.1.2 Mobilgeräte und soziale Medien als Sensorik

Mit der nahezu flächendeckenden Verbreitung von Mobilgeräten und sozialen Medien (beispielsweise Twitter und Facebook) in der Bevölkerung werden mehrere neue Anwendungsszenarien für Frühwarnsysteme und das Katastrophenmanagement ermöglicht und erforscht:

1. Integration der in einem Mobilgerät vorhandenen Sensoren in Sensorsysteme [Meissen, 2014a], [Kong, 2016]
2. Nutzung der Mobilgeräte zur (manuellen) Erfassung von Messungen und Beobachtungen [Meissen, 2014a]
3. Akquisition von Katastrophenhelfern [Meissen, 2014a]
4. Auswertung sozialer Medien zur Gefahrenerkennung und Situationsanalyse [Temnikova, 2015], [Sakaki, 2010], [Hilbring, 2016]
5. Verbreitung von Warnmeldungen über mobile Endgeräte

In internationale Publikationen wird für die Punkte 1 und 2 der englische Begriff *Human Sensors* und für Punkt 3 *Crowdsourcing* verwendet [Meissen, 2014a], [Kong, 2016].

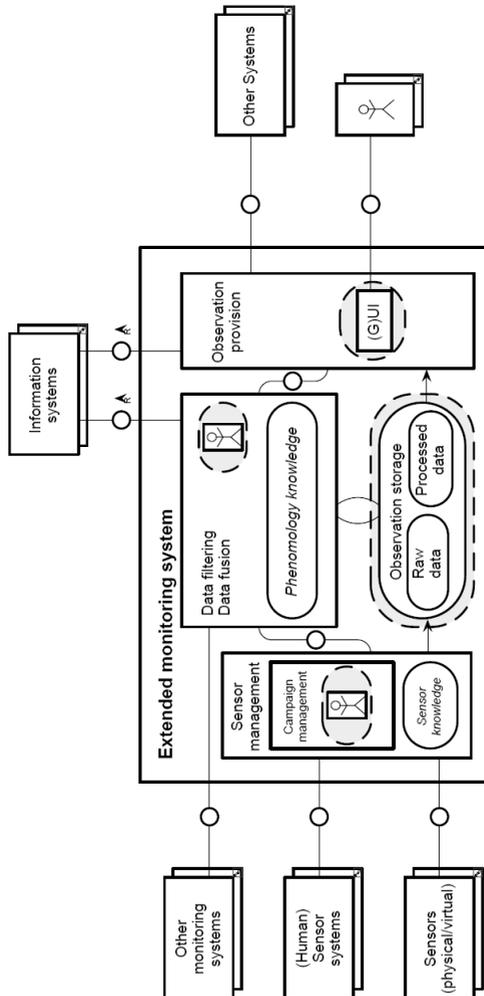


Abbildung 42: Architektur zur generischen Integration von Human Sensors [Meissen, 2014a]

Für die generische Integration von *Human Sensors* schlagen Meissen und Fuchs-Kittowski die in Abbildung 42 gezeigte Architektur vor [Meissen, 2014a]. Die Architektur erweitert den *Tsunami Sensor Bus* der GITEWS Architektur (s. Abschnitt 3.4.3.1) um den neuen Sensortyp Human Sensors. Außerdem wird ein Modul zum Kampagnenmanagement (um obigen Punkt 3 zu adressieren) innerhalb des Sensormanagements ergänzt. Wie dieses ausgestaltet werden soll, erläutern die Autoren jedoch nicht.

3.4.1.3 Verbreitung von Warnungen

Mit der zunehmenden Verbreitung neuer Informations- und Kommunikationstechnologien in der Bevölkerung in den letzten Jahren, insbesondere Web-Technologien, mobile Geräte und individuelle Nachrichtendienste, besteht nun eine Grundlage für die Entwicklung neuer personalisierter mobiler Warnsysteme. Die ersten Schritte in Richtung dieser Art der gezielten Alarmierung wurden bereits vor einiger Zeit von Valtonen et al. [Valtonen, 2004] und Addams-Moring et al. [Addams-Moring, 2005] unternommen. Diese Systeme arbeiten bei der Nachrichtenfilterung mit statischen Profilen und die Individualisierung findet vollständig auf einem zentralen Server statt.

Spätere Systeme verwenden die Position des Mobilgeräts als dynamischen Parameter und bieten die ersten Schritte zu einem Kontextbewusstsein. Allerdings stellen diese Lösungen kein allgemeines Modell zur Verfügung für die Verarbeitung von Empfängerprofilen und dem Kontext für die Alarmierung [Meissen, 2013]. Zur Personalisierung der Alarmierung stellen Malizia et al., eine Ontologie vor. Diese hat den Fokus, die Zugänglichkeit von Warnungen für beeinträchtigte Menschen zu verbessern [Malizia, 2010]. Montanari et al. entwickelten eine logikbasierte Sprache, um mit einem regelbasierte Ansatz Warnungen an die verschiedenen Teile der Bevölkerung zu versenden abhängig von ihrer aktuellen Situation [Montanari, 2007].

Ein wenig untersuchter Aspekt ist die Generierung mehrsprachiger Meldungen bei länderübergreifenden Warnsystemen. Hier wird meist nur mit statischen vorkonfigurierten Texten gearbeitet. Bouayad-Agha zeigt eine Möglichkeit auf, wie dies dynamisch mit der Hilfe von Ontologien gelöst werden kann [Bouayad-Agha, 2012].

Stellvertretend für die vielfältigen Konzepte zur Verbreitung von Warnmeldungen wird im folgenden Abschnitt PWS vorgestellt.

Policy-driven Warning System (PWS)

Montanari et al. gliedern die Verbreitung von Warnmeldungen in vier Komponenten (s. Abbildung 43). Die ersten drei Komponenten arbeiten mit einer Ontologie, der durch hinterlegte Regeln die Daten für den nächsten Schritt hinzugefügt bzw. gefiltert werden [Montanari, 2007].

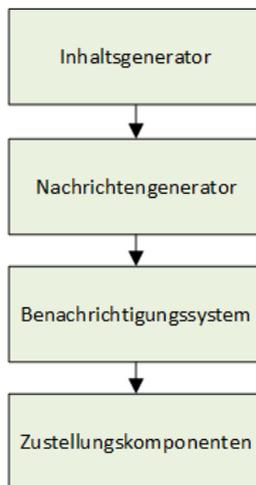


Abbildung 43: Architektur zur Verbreitung von Warnmeldungen nach [Montanari, 2007]

Inhaltsgenerator: Die erste Schicht unterstützt die automatische oder manuelle Erstellung von Meldungen. Als Eingabe dienen Warnsignale von Sensoren, externen Systemen oder die Eingabe erfolgt direkt durch die Betreiber des Systems.

Nachrichtengenerator: Diese Komponente verwendet das lokale Wissen über den geographischen Bereich, der betroffen ist und kann damit die zu informierenden Organisationen identifizieren. Dieses Wissen ist in einer GIS Datenbank gespeichert. Sind die Empfänger bekannt, werden Meldungen aus entsprechenden Vorlagen erzeugt, die die personalisierten Profile berücksichtigen.

Benachrichtigungssystem: Diese Ebene wählt die konkreten zu informierenden Personen der jeweiligen Organisation aus und auf welche Art diese benachrichtigt werden sollen. Die Personen ergeben sich aus dem hinterlegten Notfallplan der jeweiligen Organisation, in welchem die Entscheider dokumentiert sind.

Zustellungskomponenten: Für jede unterstützte Zustellungsart ist ein entsprechendes Modul vorhanden, beispielsweise SMS, mobile App, E-Mail und Fax.

Tabelle 17 fasst die Vor- und Nachteile der PWS Architektur zusammen.

Tabelle 17: Bewertung PWS

Vorteile	Personalisierte und zielgerichtete Generierung und Zustellung von Warnmeldungen.
Nachteile	Die Erstellung mehrsprachiger Meldungen als Personalisierungsoption wird nicht unterstützt.

3.4.2 Erdbeben

Stellvertretend für die zahlreichen Frühwarnsysteme für Erdbeben (zum Beispiel [Wenzel, 2014], [Rainieri, 2010], [Sakaki, 2010] oder [Hilbring, 2014]) wird das System PRESTo vorgestellt.

3.4.2.1 PRESTo

PRESTo (Probabilistic and Evolutionary early warning System) ist das Frühwarnsystem für Süditalien [Satriano, 2011], [Wenzel, 2014]. Die Systemarchitektur ist in Abbildung 44 dargestellt und folgt dem in Abschnitt 3.4.1 beschriebenen generischen Ablauf von Sensorik, Datenerfassungsschicht, Informationsverarbeitungsschicht und Informationsverbreitungsschicht. Die Informationsverarbeitungsschicht untergliedert sich in vier Berechnungsmodule, die spezifisch für Erdbeben sind:

1. Erkennung des Erdbebens (engl. Event Detection)
2. Berechnung des Ursprungsortes des Erdbebens (engl. Event Location)
3. Berechnung der Stärke des Erdbebens (engl. Magnitude Estimation)
4. Berechnung der Bodenbewegung (engl. Ground Motion)

Tabelle 18 fasst die Vor- und Nachteile der PRESTo Architektur zusammen.

Tabelle 18: Bewertung PRESTo

Vorteile	Klassische Architektur für Frühwarnsysteme mit Fokus auf Erdbeben.
Nachteile	Das System ist intern zwar modular aufgebaut, jedoch als Gesamtsystem monolithisch. Die Verteilung erfolgt rein über das angeschlossene Sensornetzwerk.

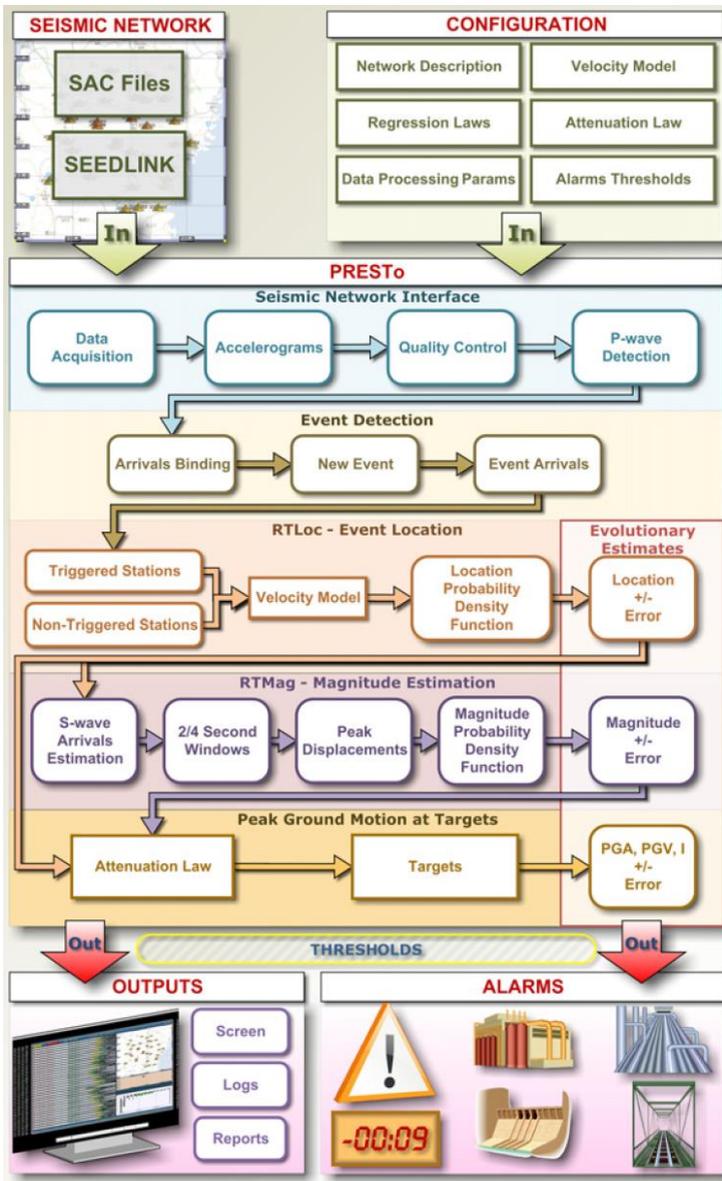


Abbildung 44: PRESto Architektur und Datenfluss [Satriano, 2011]

3.4.3 Tsunami

Eine Tsunamiwelle entsteht aufgrund eines maritimen Erdbebens. Ein Tsunami-Frühwarnsystem (TEWS, engl. Tsunami Early Warning System) muss deshalb ein Erdbebenwarnsystem enthalten.

3.4.3.1 GITEWS und DEWS

Im Jahre 2005 wurde das Projekt German Indonesian Tsunami Early Warning System (GITEWS) unter Federführung des Deutschen GeoForschungs-Zentrums (GFZ) gestartet. Das Bundesministerium für Bildung und Forschung (BMBF) finanzierte dieses Projekt als Folge des Sumatra Erdbebens 2004 zur Unterstützung der indonesischen Regierung. Im Rahmen von GITEWS wurden Sensoren (Bojen mit Unterwasserdrucksensoren, Gezeitsensoren, GPS-Sensoren und Seismometer) entwickelt und im Bereich des Indischen Ozeans installiert [Rudloff, 2006]. Aufbauend auf dieser Sensorik wurde im Rahmen des EU Projekts Distance Early Warning System (DEWS) unter Beteiligung der Länder Indonesien, Thailand und Sri Lanka ein Frühwarnsystem aufgebaut [Wächter, 2009].

Die Grobarchitektur des Systems ist in Abbildung 45 gezeigt und besteht aus vier Hauptkomponenten: Sensorik (engl. Sensors), Datenintegration (engl. Data Integration), Entscheidungsunterstützungssystem (engl. Decision Support System) und die Verbreitung der Frühwarnung (engl. Dissemination). GITEWS verwendet maritime, terrestrische und Küstensensoren zur Messung seismischer Aktivitäten, instabiler Verschiebungen und der Überwachung des Meeresspiegels. Diese Informationen werden gesammelt und in eine Datenbasis integriert. Das Entscheidungsunterstützungssystem vergleicht die Ist-Situation mit vorausgerechneten Szenarien und schlägt ggf. Alarm. Die Verbreitung der Warnmeldungen erfolgt durch den indonesischen Wetterdienst, lokale Medien und Behörden [Pinto, 2014].

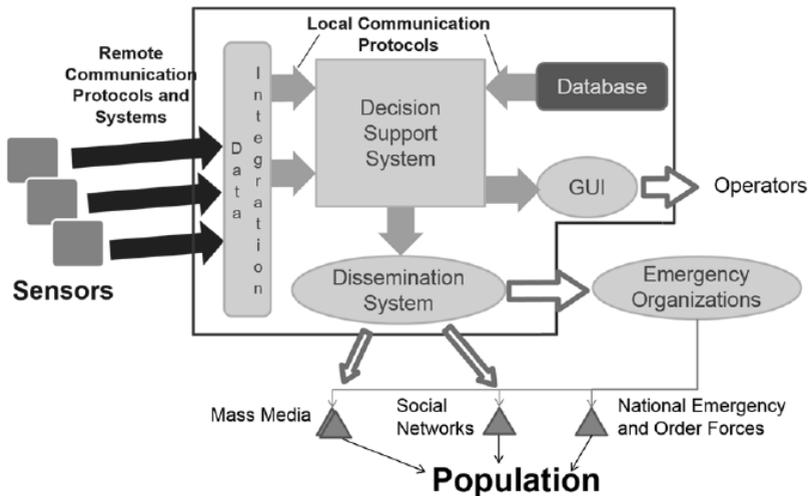


Abbildung 45: GITEWS / DEWS Grobarchitektur [Pinto, 2014]

Die Systemarchitektur von GITEWS setzt auf OGC SWE auf. Die in Abbildung 46 aufgeführten OGC Dienste SOS, SAS, WNS, SPS und Registry spielen im System dabei, wie in Abschnitt 3.2.2 erklärt, zusammen. Als Erweiterung der SWE Architektur wurde der sogenannte *Tsunami Sensor Bus (TSB)* eingeführt [Fleischer, 2010]. Der TSB soll mehrere Architekturprobleme adressieren:

- Entkopplung (lose-Kopplung) der Sensorik durch eine EDA-Schicht
- Integration der Daten in ein einheitliches Datenmodell
- Plugin-Konzept für neue Sensorik

Zur Umsetzung der EDA wurde eine MOM integriert, diese verwendet den *Java Message Service (JMS)*⁶. Zur Anbindung eines neuen Sensortyps muss ein Adapter um den Sensor entwickelt werden, der die Anbindung an JMS übernimmt. Weiterhin muss ein Plugin für den TSB implementiert werden, welches die Integration der spezifischen Daten des Sensors in das O&M

⁶ <https://java.net/projects/jms-spec/pages/Home/>, letzter Abruf am 4.9.2016

Modell des TSBs übernimmt. Tabelle 19 fasst die Vor- und Nachteile der GITEWS/DEWS Architektur zusammen.

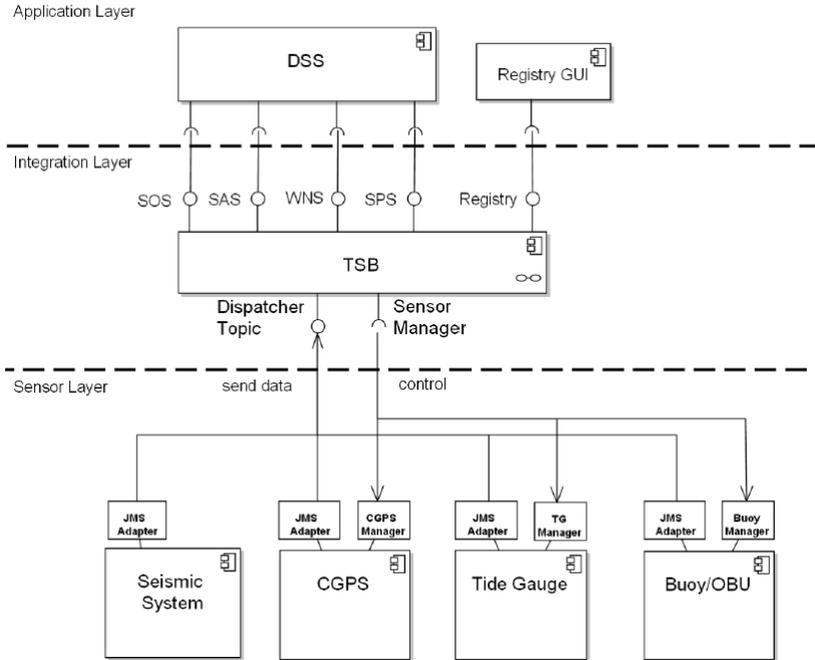


Abbildung 46: GITEWS Architektur mit Tsunami Sensor Bus (TSB) [Fleischer, 2010]

Tabelle 19: Bewertung GITEWS/DEWS

Vorteile	GITEWS/DEWS erweitert die Lösungen der SWE-Architektur um ein Konzept zur lose-gekoppelten Integration von Sensorik. Die Vorteile von SWE und EDA/MOM wurden in den entsprechenden Abschnitten bereits aufgeführt.
Nachteile	Als Erweiterung der SWE-Architektur bringt die GITEWS/DEWS-Architektur die gleichen Nachteile (s.o.) mit sich. Das gute Konzept des TSB besitzt Nachteile, falls sich die Sensorik regelmäßig ändert (zum Beispiel durch neue und bessere auf dem Markt verfügbare Sensoren). Hierzu sind, wie oben erklärt, mehrere Implementierungen notwendig und der Austausch kann nicht dynamisch durch Konfiguration erfolgen.

3.4.3.2 TEWS Architecture

Pinto und Cipriano nehmen in ihrer Veröffentlichung *Tsunami Early Warning System Architecture based on Automation Technologies* eine theoretische Untersuchung vor, die die Vorteile des Einsatzes industrieller Steuersysteme (SCADA, engl. Supervisory Control and Data Acquisition) für die Umsetzung eines Systems zur Tsunami-Frühwarnung betrachtet [Pinto, 2014]. Sie führen folgende Argumente für ihre Idee auf:

- Eine Architektur, bestehend aus kommerziellen Komponenten, ist robuster, da die Komponenten bereits in anderen Systemen im Einsatz und damit erprobt sind. Dies vermeidet außerdem den Aufwand zur Fehlersuche bei Neuentwicklungen.
- Produkte müssen nicht neu entwickelt, sondern nur integriert werden.
- Die Produkte sind für den Einsatz im harten industriellen Umfeld entworfen und somit per se robust.

Die Autoren gliedern ein TEWS in 4 Teilsysteme: Sensorik, Kommunikation, Verarbeitung und Verbreitung der Frühwarnung. Für alle Teilsysteme werden verfügbare Komponenten vorgeschlagen, der Integrationsaufwand wird jedoch nicht betrachtet.

Tabelle 20 fasst die Vor- und Nachteile der TEWS Architektur zusammen.

Tabelle 20: Bewertung TEWS

Vorteile	Nutzung bewährter Technologien aus dem Industrieumfeld.
Nachteile	<p>Für TEWS werden Sensoren benötigt, die nicht im industriellen Umfeld verfügbar sind. D.h. auch die Integration in ein SCADA System muss für diese Spezialsensorik gelöst werden.</p> <p>Die industriellen Kommunikationsnetze sind zwar auf widrige Umstände ausgelegt (zum Beispiel in einem Stahlwerk), jedoch nicht für den Katastrophenfall.</p> <p>SCADA Systeme können besonders gut Daten einsammeln (beispielsweise Prozesswerte und Stückzahlen), jedoch ist die Datenintegration immer noch nicht semantisch gelöst. Dies könnte sich jedoch mit der zunehmenden Digitalisierung der Industrie ändern, beispielsweise durch die Initiative Industrie 4.0 [Bauernhansl, 2014]. Diese Diskussion würde in dieser Arbeit zu weit führen, hierzu sei verwiesen auf die Arbeit von Schleipen [Schleipen, 2012].</p>

3.4.4 Flut

Architekturen zu Flutfrühwarnsystemen findet man mehrere in der Literatur, die jedoch aus Sicht der Systemarchitektur keine neuen Erkenntnisse gegenüber den bereits vorgestellten Architekturen geben. Shi et al. beschreiben beispielsweise ein Flutwarnsystem basierend auf einer SOA [Shi, 2015] und Sharma et al. basierend auf einer Event-Driven SOA [Sharma, 2016]. Neuere Forschungsansätze findet man in der folgenden CIS-Architektur.

3.4.4.1 Flood EWS und die CIS Architektur

Das Flutfrühwarnsystem *Flood EWS* [Pengel, 2013] nutzt In-situ-Sensoren zur Überwachung von Böschungen in städtischen Gebieten, um die

Vorhersage eines Deichbruches und die daraus resultierenden Auswirkungen zu berechnen.

Der hierzu notwendige Ablauf im Flood EWS ist in Abbildung 47 dargestellt. Im Standardmodus werden die Sensordaten kontinuierlich durch das Sensor Anomaly Detection Modul analysiert, das Algorithmen des maschinellen Lernens für die Erkennung von Anomalien in Sensorsignalen nutzt [Pengel, 2013].

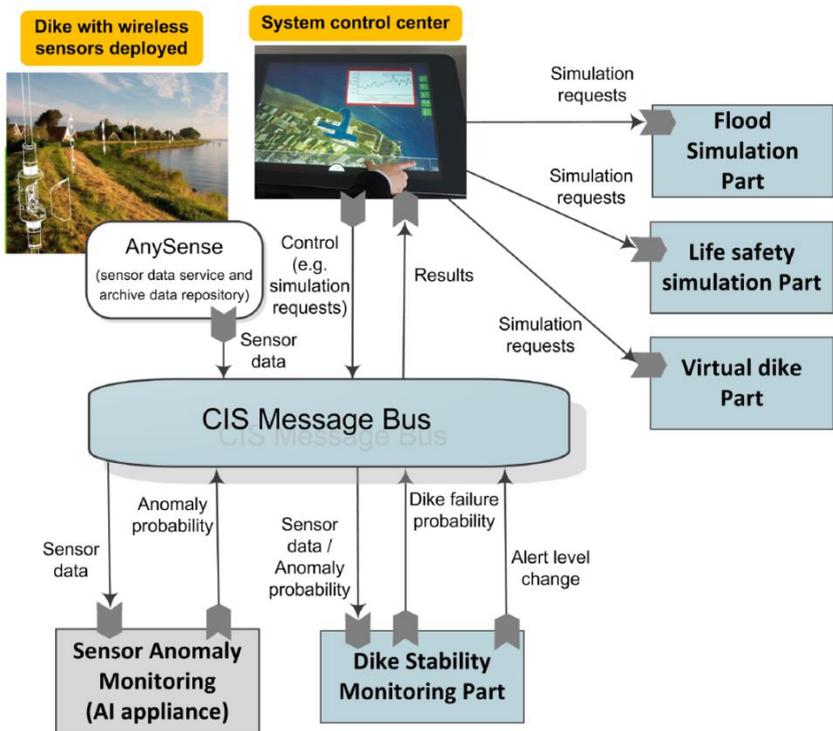


Abbildung 47: Flutwarnsystem basierend auf CIS [Balis, 2011]

Wenn die Wahrscheinlichkeit einer Anomalie einen hohen Wert erreicht, wird automatisch die Berechnung des Risikos eines Deichbruchs gestartet (engl. Dike Stability Monitoring). Falls auch diese Berechnung ein hohes Risiko ergibt, können zwei Simulationen durchgeführt werden: Die erste umfasst die Berechnung der Überflutung des bedrohten Gebiets basierend auf der möglichen Bruchstelle (engl. Flood Simulation Part), die zweite prognostiziert Evakuierungsmuster und den Verlust von Menschenleben im Falle einer Überschwemmung (engl. Life Safety Simulation Part). Ein separates Werkzeug ist der Virtual Dike Simulator, der das Rechenmodell eines Deiches implementiert und für fortgeschrittene Experimente verwendet werden kann, um beispielsweise verschiedene Szenarien durchzuspielen.

Flood EWS basiert auf der Common Information Space Plattform (CIS) die im EU FP7 Projekt UrbanFlood entwickelt wurde [Balis, 2011]. Testinstallationen von Flood EWS wurden während des Projekts in Großbritannien (Bostondeich) und in den Niederlanden (Stammerdijk, LiveDijk, Rheindeich) durchgeführt.

CIS berücksichtigt die INSPIRE⁷ Richtlinien für Geodaten. Weiterhin werden die Berechnungsalgorithmen und Simulatoren (in CIS *Parts* genannt) als OGC Web Processing Service gekapselt. Zur Orchestrierung der *Parts* werden diese über eine MOM verbunden (*CIS Message Bus*). Zur Umsetzung von Szenarien können diese mit anderen Diensten verbunden werden, hierzu wird Apache Camel⁸ als Integrationsplattform eingesetzt. Als interessanter Aspekt wurde im Projekt am Betrieb der Komponenten in einer Cloud-Infrastruktur gearbeitet.

Tabelle 21 fasst die Vor- und Nachteile der Flood EWS Architektur zusammen.

⁷ <http://inspire.ec.europa.eu/>, letzter Abruf am 4.9.2016

⁸ <http://camel.apache.org/>, letzter Abruf am 4.9.2016

Tabelle 21: Bewertung Flood EWS

Vorteile	Das Projekt verwendet ähnliche Konzepte wie im Projekt SANY, jedoch wird zur Entkopplung der Berechnungsprozesse eine MOM eingesetzt. Die Prozesse können über eine Cloud skaliert werden.
Nachteile	Zur Orchestrierung der Dienste wird kein BPMS integriert, sondern nur eine Integration der Dienste auf Ebene der Kommunikationsschicht durchgeführt.

3.5 Zusammenfassung

Wie die Analyse existierender Architekturen gezeigt hat, besitzen diese noch viele Einschränkungen und sind nicht für die Anforderungen zukünftiger Frühwarnsysteme ausgelegt. Die Thematik der Resilienz wird beispielsweise noch in keinem Ansatz berücksichtigt. Den Einsatz von BPMS in Frühwarnsystemen zur Umsetzung von SOPs findet man in der Literatur ebenfalls nicht. Klassische Orchestrierungen einer SOA werden dagegen öfter beschrieben, beispielsweise im europäischen Projekte ORCHESTRA [Usländer, 2007]. In SANY (s. Abschnitt 3.2.3) wurde BPEL eingesetzt, um die Reihenfolge der Analyse der Sensordaten zu implementieren.

Bevor jedoch eine vertiefende Bewertung erfolgen kann, müssen zunächst die Anforderungen an ein Frühwarnsystem der nächsten Generation aufgestellt werden. Dies erfolgt im folgenden Kapitel.

4 Anforderungen an Frühwarnsysteme der nächsten Generation

In den folgenden Abschnitten werden die Anforderungen an ein Rahmenwerk für Frühwarnsysteme der nächsten Generation (RA-FWS^{NG}) analysiert. Diese werden aus den unterschiedlichen Bereichen, die in das Forschungsgebiet der Frühwarnsysteme einfließen (s. Kapitel 2), abgeleitet. Dies sind u.a. Anforderungen an Entscheidungsunterstützungssysteme und an verteilte Systeme. Ergänzend werden die neuen Anforderungen aus der Resilienz betrachtet. Das Kapitel schließt mit einer Zusammenfassung der Anforderungen in Abschnitt 4.5. Die dort fortlaufend vergebenen Nummern werden bereits in den folgenden Texten in Klammer angegeben (FA-X und NFA-X, wobei X eine Zahl ist).

4.1 Anforderungen an Frühwarnsysteme

Wie in Abschnitt 2.1 beschrieben wurde, setzt sich die Frühwarnung aus drei Punkten zusammen. Zusätzlich ist als Schnittstelle zum übergreifenden Katastrophenmanagement als vierter Punkt die Verbesserung und Unterstützung der Reaktionsfähigkeit aufgeführt.

1. Überwachung von Indikatoren (Vorbote)
2. Prognose eines wahrscheinlichen Ereignisses
3. Kommunikation und Verbreitung von Warnungen
4. Verbesserung und Unterstützung der Reaktionsfähigkeit

Die folgenden Abschnitte sind nach diesen Punkten untergliedert.

4.1.1 Überwachung von Indikatoren

4.1.1.1 Datenquellen für Indikatoren

Frühwarnung basiert im Wesentlichen auf der Sammlung großer Datenmengen, ihrer Interpretation in Echtzeit und ihrer Visualisierung. Deshalb basieren Frühwarnsysteme auf Systemen, die Messwerte aus nicht homogenen Datenquellen sammeln [Ionita, 2008]. Ionita und Olteanu präsentieren als Ergebnis einer Literaturstudie, welche Datenquellen für welche Art der Frühwarnung notwendig sind [Ionita, 2014]. Diese Übersicht ist in Tabelle 22 dargestellt. Neben diesen Datenquellen werden auch andere Methoden als sinnvoll erachtet, wie das *Crowdsourcing*, und Daten anderer Informationssysteme (**FA-3**), beispielsweise Umweltinformationssysteme [Meissen, 2014a].

Tabelle 22: Datenquellen für Katastrophen [Ionita, 2014]

	Chemisch / Nukleare Unfälle	Erdbeben	Erdrutsche	Tsunamis	Vulkanausbrüche	Überschwemmungen	Luftverschmutzung	Wasserverschmutzung
Windmesser	X						X	X
Multispektrale Wolkenanalyse mit Satelliten	X						X	
Geologische Karten		X	X					
Geostationäre Satellitendaten		X			X	X		
Global Positioning System (GPS)		X	X		X			
Historische Katastrophenkarten		X	X		X	X	X	X
Hydrologische Karten		X	X			X		X

	Chemisch / Nukleare Unfälle	Erdbeben	Erdrutsche	Tsunamis	Vulkanausbrüche	Überschwemmungen	Luftverschmutzung	Wasserverschmutzung
Hygrometer (Luftfeuchtigkeit)	X					X		
Infrarotbilder					X			
Strahlungsmessung von ultraviolettem Licht							X	
Demographische Daten		X			X		X	
Regen / Schneemessgeräte, Mikrowellensensoren	X					X		X
Tiefseemessung und Meeresspiegel				X				
Seismische Zonenkarten	X	X	X	X				
Spektrometer und Infrarotsensoren für Gasemissionen					X			
Spektroradiometrie für Aerosole							X	
Dehnungsmesser		X				X		
Synthetic Aperture Radar (SAR)		X	X		X			
Thermische Infrarotbilder			X		X			
Thermometer						X		
Topographische und Höhen Karten	X	X	X	X	X			
Trübungsmessgeräte								X
Messung von Turbulenzen	X							

Aus dieser großen Menge an Sensorik (**FA-1, FA-4**) ergibt sich automatisch die Anforderung an die Systemarchitektur, eine dynamische

Sensorverwaltung vorzusehen (**FA-2**, **FA-5**). Weiterhin müssen die heterogenen Daten in ein gemeinsames Datenmodell überführt werden, können bzw. mit semantischen Annotationen zu einem gemeinsamen Verständnis der Inhalte führen (**FA-6**).

4.1.1.2 Berechnung von Indikatoren

Die Berechnung der Indikatoren ist eng mit der Prognose des wahrscheinlichen Ereignisses verbunden und nicht immer zu trennen. Deshalb werden diese Berechnungen in der Auflistung der Verarbeitungskategorien im nächsten Abschnitt mit aufgeführt.

4.1.2 Prognose eines wahrscheinlichen Ereignisses

Unter diesen Punkt fallen im Wesentlichen diverse Berechnungsalgorithmen. Ionita und Olteanu [Ionita, 2014] identifizierten die folgenden Arten von Methoden:

- Schätzung der abgeleiteten physikalischen Größen (zum Beispiel seismische Parameter wie Größe und Lage, Spitzen der Bodenbeschleunigung und die spektrale Beschleunigung im Falle von Erdbeben, Index der Luftqualität)
- Vorhersagen, basierend auf mathematischen Modellen (zum Beispiel Bewegung von chemischen oder radioaktiven Verunreinigungen in der Atmosphäre, Ausbreitung eines Schadstoffs im Wasser durch Ausbreitungsmodelle, Muster der Brandausbreitung) und Vorboten von Ereignissen (zum Beispiel Messwerte von Gasen und visuelle Eigenschaften als Vorboten von Vulkanausbrüchen).
- Risikobeurteilung (zum Beispiel Schätzung einer Tsunami-Entstehung in Bezug auf die Lage des Erdbebens, seiner Tiefe und Größe; mögliche Brandgefahr in Bezug auf den Zustand der Vegetation)
- Risikominimierung (zum Beispiel Entscheidungsunterstützung durch Bestimmung von Alternativen, Katastrophenmanagement)

- Organisieren und Speichern von Daten für historische Zwecke, beispielsweise für tieferegehende Analysen, Durchführung von Übungen
- Erzeugung von Daten für Kommunikationszwecke, die als Basis für die Verbreitung von Warnungen dienen

Ähnlich wie bei der Verwaltung von Sensorik gibt es eine Vielzahl von Algorithmen, die verwaltet werden müssen (**FA-7**). Durch neue Erkenntnisse entstehen neue Algorithmen und müssen dynamisch ausgetauscht werden können (**FA-8**).

4.1.3 Kommunikation und Verbreitung von Warnungen

In der Kommunikation sind nicht nur die Endempfänger in der Bevölkerung zu betrachten, sondern alle Akteure, die zur Entscheidungsfindung beitragen, beispielsweise Fachexperten. Diese entscheiden je nach fachlicher Ausrichtung in enger Abstimmung miteinander, ob der Alarm berechtigt ist, weitere Untersuchungen notwendig sind oder ob ein Fehlalarm vorliegt [Bell, 2010]. Dies kann auch einen mehrstufigen Prozess erfordern. Diese Prozesse sind meist in SOPs (s. Abschnitt 2.1.1.2) beschrieben und müssen flexibel in ein Frühwarnsystem integriert und modifiziert werden können (**FA-10**).

Scholte und Rothkrantz analysierten mehrere Veröffentlichungen zu den Anforderungen an den Benachrichtigungsprozess, diese wurden aggregiert zu den folgenden Anforderungen [Scholte, 2014]:

- *Personalisierung*: Der Inhalt und die Formulierung müssen auf den Empfänger zugeschnitten sein.
- *Kontextsensitiv*: Der Alarm sollte eine Erklärung der aktuellen Situation enthalten.
- *Adaptiv*: Die Warnung enthält Verhaltensempfehlungen.

- *Auswertung und Dynamik*: Im Idealfall bestätigt der Empfänger die Nachricht und die Aktualisierungen der Meldungen sind an die aktualisierte Situation angepasst.

An die Systemarchitektur ergibt sich daraus die Anforderung, personalisierte und kontextsensitive Nachrichten generieren zu können, die auf den aktuellen Situationsdaten basieren (**FA-12**). Als Parameter der Personalisierung sind die Sprache des Empfängers (**FA-13**) und der gewünschte Kommunikationskanal zu berücksichtigen [IFRC, 2012]. Weiterhin müssen alternative Kommunikationswege zur Verfügung stehen (**FA-14**) für den Fall, dass die gewünschte Kommunikationsart ausfällt [IFRC, 2012].

4.1.4 Reaktionsfähigkeit

Die Entwicklung und Umsetzung eines wirksamen Frühwarnsystems erfordern den Beitrag und die Koordinierung eines breiten Spektrums von Einzelpersonen und Gruppen. Die möglichen Arten von Organisationen und Gruppen, die in die Funktionen und Aufgaben eines Frühwarnsystems einbezogen werden sollten, können sich ändern. Die identifizierten Beteiligten müssen in die im vorherigen Abschnitt genannten SOPs integriert und dynamisch angepasst werden können (**FA-16**).

Karten eignen sich in den verschiedensten Phasen der Frühwarnung zur Visualisierung von Gefahren, Sensorstandorten, Simulationsergebnissen, etc. und verbessern damit auch das Verständnis der aktuellen Lage. Zur Implementierung von Karten kommen geographische Informationssysteme zum Einsatz (**FA-11**) und sind damit ein essentieller Bestandteil von Frühwarnsystemen auf lokaler, nationaler und regionaler Ebene [IFRC, 2012].

4.2 Anforderungen zur Entscheidungsunterstützung

Entscheidungsunterstützung ist ein breiter Begriff, der auf jeden Mechanismus angewandt werden kann, der in irgendeiner Form menschliche Entscheidungsträger unterstützt, ihre Entscheidungen besser und schneller zu treffen. Nach Power können Entscheidungsunterstützungssysteme in fünf Kategorien eingeteilt werden:

1. *Datengetriebene* (engl. Data-driven) Entscheidungsunterstützung konzentriert sich auf die Analyse großer Datensammlungen, um signifikante Muster, Sequenzen oder Cluster zu identifizieren. Dieser Prozess wird in der Regel als *Data-Mining* bezeichnet
2. *Modellgetriebene* (engl. Model-driven) Entscheidungsunterstützung verwendet Simulationen (**FA-9**), Modelle und Optimierungsmethoden, um Metriken zu berechnen (zum Beispiel Kosten, Risiko), die den Entscheidungsprozess unterstützen
3. *Wissensbasierte* (engl. Knowledge-driven) Lösungen nutzen formal dargestelltes Wissen und verwenden beispielsweise Inferenzmethoden, um Aktionen auf Basis von Domänenwissen vorzuschlagen oder zu empfehlen (**FA-15**)
4. *Kommunikationsgetriebene* (engl. Communication-driven) Systeme erleichtern die Kommunikation und Zusammenarbeit von Verantwortlichen und Fachexperten für die Entscheidungsfindung
5. *Dokumentengesteuerte* (engl. Document-driven) Entscheidungsunterstützung konzentriert sich auf intelligente Dokumentenrecherche, um hilfreiche Dokumente für den Entscheidungsprozess zur Verfügung zu stellen (**FA-15**)

Ein modernes Frühwarnsystem muss alle fünf Kategorien unterstützen und sogar integrieren. Die Anforderungen für Kategorie 1, 2 und 4 wurden bereits im Abschnitt der Anforderungen an Frühwarnsysteme abgedeckt (s. Abschnitt 4.1).

4.3 Anforderungen verteilter Systeme

In Kapitel 1 wurde gezeigt, dass Frühwarnsysteme eine Unterkategorie verteilter Systeme sind. Daraus ergeben sich besondere Herausforderungen bei der Architektur von Frühwarnsystemen, die von Moßgraber et al. in *die sieben größten Herausforderungen bei der Definition einer Architektur von Frühwarnsystemen* [Moßgraber, 2013] diskutiert wurden:

1. Erstellung einer skalierbaren Kommunikationsinfrastruktur für ein System von Systemen (s. Abschnitt 2.2.3.2)
2. Erstellung einer resilienten Kommunikationsinfrastruktur für SoS
3. Effiziente Verwaltung großer Datenmengen mit semantischen Annotationen
4. Skalierbare und schnelle Speicherung von großen und verteilten Datenmengen
5. Handhabung heterogener Daten aus unterschiedlichen Domänen
6. Verwaltung und Auffinden von Sensorik in räumlich verteilten SoS
7. Koordination der Aufgaben in räumlich verteilten SoS

Die Punkte 5 (**FA-6**), 6 (**FA-5**) und 7 (**FA-10**) wurden bereits bei den Anforderungen an Frühwarnsysteme aufgeführt (s. Abschnitt 4.1). Die Anforderungen aus den Punkten 1 bis 4 werden in den folgenden Abschnitten beschrieben. Danach wird der Sicherheitsaspekt in verteilten Systemen betrachtet.

4.3.1 Skalierbare Kommunikationsinfrastruktur für ein SoS

Verschiedene Systemkomponenten (Subsysteme) eines Frühwarnsystems, zum Beispiel ein Katalog, die Wissensbasis oder ein BPMS, können entweder am gleichen Standort vorhanden oder auf mehrere Standorte verteilt sein. Die Kommunikation zwischen diesen Standorten muss sich an die Anforderungen an Geschwindigkeit und Datenmenge anpassen können und somit skalierbar sein. Die Subsysteme eines SoS sind nicht einheitlich

entworfen und umgesetzt. Sie verwenden verschiedene Programmiersprachen (Java, C#, etc.) und sind auf verschiedenen Betriebssystemen bereitgestellt (Linux, Windows). Eine solche Heterogenität der Subsysteme erfordert eine offene und standardisierte Kommunikationsschicht (**NFA-1**).

4.3.2 Resiliente Kommunikationsinfrastruktur für SoS

Während des Austauschs von Nachrichten können Systemausfälle, Verbindungsfehler und ein Clientausfall auftreten, entweder aufgrund fehlender Systemressourcen oder einer schlechten Dienstgüte, die einen Nachrichtenverlust zur Folge hat. Beispielsweise muss die Zustellung von kritischen Nachrichten bei Tsunami-Frühwarnsystemen zwischen regionalen und nationalen Warnzentralen sichergestellt sein. Daher wird für den Nachrichtenverbreitungsprozess im SoS eine resiliente Kommunikationsschicht benötigt (**NFA-2**).

4.3.3 Effiziente Verwaltung großer Datenmengen

In Frühwarnsystemen muss mit großen Mengen heterogener Daten umgegangen werden. Dies sind zeitliche Datensätze (zum Beispiel Zeitreihen von Gezeitensensoren), räumliche Datensätze (zum Beispiel Shape-Dateien, die räumlich geclusterte Ergebnisse einer Datenfusion repräsentieren), räumlich-zeitliche Datenmengen (zum Beispiel die Ausbreitung einer Verunreinigung oder einer Tsunamiwelle), Datensätze aus sozialen Medien (zum Beispiel Twitter) und thematische Datensätze (zum Beispiel Klassifikationen von Erdbeben mit Tsunamipotential). Der Datendurchsatz kann je nach Datenquelle stark variieren (von stündlichen bis sekundlichen Messungen). Die Herausforderung besteht darin, diese Daten effizient zugänglich zu machen und zwar unter Beibehaltung der semantischen Bedeutung der Daten für eine spätere intelligente Verarbeitung (**NFA-3**).

4.3.4 Skalierbare Speicherung großer verteilter Datenmengen

Die Entstehung großer Datenmengen wurde bereits im vorherigen Abschnitt motiviert. Damit stellt sich die Herausforderung, diese verteilten Datensätze effizient zu speichern sowohl für die Verarbeitung in Echtzeit als auch für die tiefgehende Offline-Datenanalyse. Viele Verarbeitungsdienste haben die Anforderung gleichzeitig auf mehrere Datensätze zugreifen zu müssen, beispielsweise auf im Cache gespeicherte Sensordaten und auf Daten vorausberechneter Ausbreitungsmodelle (**NFA-4**).

4.3.5 Sicherheit in verteilten Systemen

In verteilten Systemen müssen die Vertraulichkeit, die Integrität und die Verfügbarkeit von Ressourcen gewährleistet werden. Angriffe auf die Sicherheit sind beispielsweise das Abhören oder die Manipulation von Daten und Denial-of-Service (DoS) [Coulouris, 2012]. Architekten sicherer verteilter Systeme müssen Vorkehrungen treffen, damit offene Dienst-schnittstellen und unsichere Netze nicht zur Kompromittierung des Systems führen (**NFA-5**).

4.4 Anforderungen aus der Resilienz

Die neuen Ergebnisse aus der Resilienzforschung erfordern neue Qualitäten zukünftiger Frühwarnsysteme. Der in Abschnitt 2.2.1.4 geforderte Umgang mit kontinuierlich auftretenden disruptiven Ereignissen erfordert ein Umdenken weg von der reinen Frühwarnung und hin zu einem *Ereignismanagementsystem* (EMS). Zunächst müssen die potentiellen Bedrohungen, Gefahren und Störungen, die kontinuierlich auftreten können, identifiziert werden [Häring, 2016]. Ein EMS muss den parallelen Umgang mit eintretenden Ereignissen ermöglichen. Dazu wird eine integrierte Benutzeroberfläche benötigt (**FA-17**) [Robinson, 2015], die den

Anwendern den notwendigen Überblick verschafft, die Verknüpfung und Aggregation von Ereignissen unterstützt (**FA-18**) und die Arbeit mit parallelen Simulationen zu den unterschiedlichen Ereignissen ermöglicht (inkl. der Integration neuer Ereignisse) (**FA-19**).

4.5 Aggregierte Anforderungen

Im Folgenden sind die oben genannten Anforderungen getrennt nach funktionalen (Tabelle 23) und nichtfunktionalen Anforderungen (Tabelle 24) aufgeführt.

Tabelle 23: Funktionale Anforderungen an RA-FWS^{NG}

Nummer	Anforderung
	Generische Sensorintegration
FA-1	- statisch
FA-2	- dynamisch
FA-3	- alternative Datenquellen (Soziale Medien, Crowdsourcing, Informationssysteme, etc.)
	Sensormanagement
FA-4	- Katalog
FA-5	- Semantischer Katalog
FA-6	Integration heterogener Sensordaten
	Generische Integration von Algorithmen
FA-7	- statisch
FA-8	- dynamisch
FA-9	Dynamische Integration von Simulationen
FA-10	Integration von Standard Operating Procedures
FA-11	Geographisches Informationssystem
	Warnmeldungen
FA-12	- personalisiert
FA-13	- mehrsprachig
FA-14	- mehrere und alternative Kommunikationswege
FA-15	Recherche der Wissensbasis und von Dokumenten

FA-16	Flexible Integration von Akteuren
FA-17	Integrierte Benutzeroberfläche
FA-18	Aggregation von Ereignissen
FA-19	Parallele Simulationen

Tabelle 24: Nichtfunktionale Anforderungen an RA-FWS^{NG}

Nummer	Anforderung
NFA-1	Erstellung einer skalierbaren Kommunikationsinfrastruktur für ein SoS
NFA-2	Erstellung einer resilienten Kommunikationsinfrastruktur für SoS
NFA-3	Effiziente Verwaltung großer Datenmengen
NFA-4	Skalierbare Speicherung großer verteilter Datenmengen
NFA-5	Sicherheit in verteilten Systemen

5 Ein Architekturrahmenwerk für Frühwarnsysteme

Die Spezifikation des Rahmenwerks für die Architektur von Frühwarnsystemen der nächsten Generation RA-FWS^{NG} erfolgt anhand von RM-ODP. RM-ODP wurde ausgewählt, da es sich für die Architektur von verteilten Systemen in zahlreichen Projekten der unterschiedlichsten Domänen (Telekommunikation, Finanzwirtschaft, Produktion, Geographische Systeme, Regierungsbehörden, etc.) bewährt hat [Putman, 2001], [Usländer, 2010], [Moßgraber, 2012a].

Die folgenden Abschnitte wurden deshalb entsprechend der von RM-ODP vorgegebenen Sichten gegliedert (s. Abschnitt 2.4.2.2) inklusive einer ergänzenden Sicht für System von Systemen. Jede Sicht schließt mit einer Liste der Schritte, die bei der Erstellung einer konkreten Architektur für ein Frühwarnsystem mit RA-FWS^{NG} durchzuführen sind.

5.1 Unternehmenssicht

In der Unternehmenssicht werden die Anforderungen der Nutzer analysiert und im Hinblick auf ihre Funktions-, Informations- und qualitativen Aspekte dokumentiert. Sie identifiziert die Hauptakteure, Rollen und Anwendungsfälle.

5.1.1 Hauptakteure

Die Entwicklung und Umsetzung eines wirksamen Frühwarnsystems erfordert den Beitrag und die Koordinierung eines breiten Spektrums von Einzelpersonen und Gruppen.

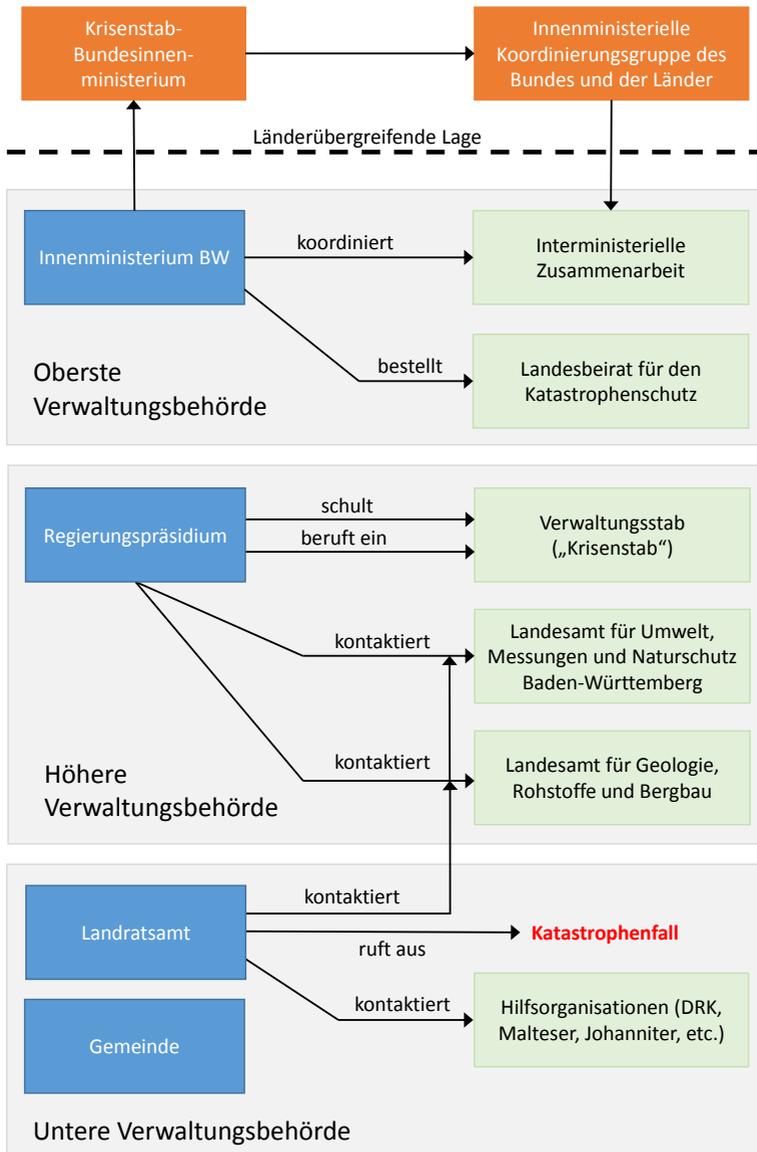


Abbildung 48: Zuständigkeiten im Katastrophenschutz in Baden-Württemberg nach [Bell, 2010]

Die folgende Liste enthält eine Beschreibung der möglichen Arten von Organisationen und Gruppen, die in die Funktionen und Aufgaben eines Frühwarnsystems einbezogen werden sollten. Bei der konkreten Erstellung eines neuen Frühwarnsystems muss immer eine Analyse der real vorhandenen Beteiligten durchgeführt werden. Ein Beispiel, zeigt Abbildung 48, die im Projekt *Integrative Frühwarnsysteme für gravitative Massenbewegungen (ILEWS)* durchgeführt wurde [Bell, 2010].

5.1.1.1 Gemeinschaften

Gemeinschaften, insbesondere die am meisten gefährdeten, sind von grundlegender Bedeutung für den Erfolg eines Frühwarnsystems. Die betroffenen Menschen werden in den Mittelpunkt der Frühwarnsysteme gerückt und sie sollen aktiv an allen Aspekten der Errichtung und des Betriebs beteiligt werden. Sie müssen sich der Gefahren und möglichen Auswirkungen, denen sie ausgesetzt sind, bewusst sein und können somit Maßnahmen ergreifen, um die Gefahr von Verlust oder Schaden zu minimieren [UNISDR, 2006].

5.1.1.2 Lokale Regierungen

Lokale Regierungen, wie beispielsweise Stadt-, Gemeinde- und Kreisverwaltungen, spielen eine wichtige Rolle bei effektiven Frühwarnsystemen, da sie ein beachtliches Wissen über die Gefahren, denen ihre Gemeinden und Bürger ausgesetzt sind, besitzen. Sie sollten von den nationalen Regierungen die entsprechenden Befugnisse erhalten, um sich aktiv an der Gestaltung und der Pflege von Frühwarnsystemen beteiligen zu können. Sie müssen die Informationen, die sie erhalten, richtig verstehen können und in der Lage sein, diese entsprechend an die lokale Bevölkerung weiterzugeben [UNISDR, 2006].

5.1.1.3 Nationale Regierungen

Nationale Regierungen sind verantwortlich für die High-Level-Strategien und Vorgaben, um die Frühwarnung zu ermöglichen. Weiterhin sind sie für die

Umsetzung technischer Vorhersagesysteme zuständig, die Katastrophenvorwarnungen auf nationaler Ebene erstellen und diese verteilen. Die nationalen Regierungen sollten sowohl mit regionalen als auch internationalen Regierungen und Behörden zusammenarbeiten, um die Frühwarnkapazitäten zu stärken und sicherzustellen, dass Warnungen zu den gefährdeten Bevölkerungsgruppen gelangen und somit die notwendigen Reaktionsmaßnahmen erfolgen können [UNISDR, 2006].

5.1.1.4 Länderübergreifende Einrichtungen

Länderübergreifende Institutionen und Organisationen spielen eine Rolle bei der Bereitstellung von Fachwissen und der Beratung. Sie unterstützen die nationalen Anstrengungen bei der Entwicklung von Frühwarnfähigkeiten in Ländern, die eine gemeinsame geographische Umgebung teilen. Darüber hinaus fördern sie die Verbindungen zu internationalen Organisationen und erleichtern den Austausch effektiver Früherkennungspraktiken zwischen benachbarten Ländern [UNISDR, 2006].

5.1.1.5 Internationale Einrichtungen

Internationale Gremien können für die Verbesserung der länderübergreifenden Koordinierung sorgen, die Standardisierung und Unterstützung für nationale Frühwarnaktivitäten vorantreiben und den Austausch von Daten und Wissen zwischen den einzelnen Ländern und Regionen fördern [UNISDR, 2006].

5.1.1.6 Nichtstaatliche Organisationen

Nichtstaatlichen Organisationen (engl. NGO, Non-Governmental Organisations) fällt die Rolle zu, bei der Verbesserung des notwendigen Gefahrenbewusstseins der Bevölkerung insbesondere auf kommunaler Ebene mitzuwirken. Darüber hinaus sorgen sie durch gezielte Lobbyarbeit dafür, dass das Thema der Frühwarnung auf der Tagesordnung der politischen Entscheidungsträger bleibt [UNISDR, 2006].

5.1.1.7 Firmen

Industriefirmen spielen eine vielfältige Rolle bei der Frühwarnung. Viele Firmen, wie zum Beispiel Chemiefirmen, müssen eigene Frühwarnfähigkeiten entwickeln. Hier liegt ein großes ungenutztes Potenzial, das im Krisenfall helfen kann, qualifizierte Dienstleistungen in Form von technischem Personal (zum Beispiel Werksfeuerwehren), Know-how oder Sach- und Geldspenden zur Verfügung zu stellen [UNISDR, 2006].

5.1.1.8 Wissenschaftliche Einrichtungen

Das Know-how von Wissenschaftlern ist von zentraler Bedeutung für die Analyse von natürlichen Risiken, für die Gestaltung der systematischen Datenanalyse und der Warndienste, das Übersetzen wissenschaftlicher oder technischer Informationen in verständliche Meldungen und deren Verbreitung an die jeweiligen Zielgruppen [UNISDR, 2006].

5.1.1.9 Akteure bei der Kommunikation der Warnung

Damit eine erfolgreiche Kommunikation stattfinden kann, werden mindestens drei Akteure benötigt [IFRC, 2012].

1. *Autor*: Verantwortlich für die Erstellung oder Zusammenstellung des Inhalts der Alarmnachricht (in der Regel eine Institution wie Meteorologie-, Hydrologie- oder Gesundheitsämter).
2. *Mediator*: Er aggregiert erhaltene Informationen und formuliert diese in der Sprache der weiteren Empfänger (falls dies nicht automatisiert möglich ist).
3. *Empfänger*: Der Adressat der Alarmmeldung bzw. die breitere Öffentlichkeit.

5.1.2 Rollen im Katastrophenfall

Die wichtigste Funktion für die Reaktionsfähigkeit im eingetretenen Katastrophenfall ist das Ressourcenmanagement. Einsatzkräfte müssen

effektiv eingesetzt werden und die Grundversorgung (Trinkwasser, Nahrungsmittel) muss organisiert werden. Hierzu müssen Personen für die folgenden sieben Rollen explizit bekannt sein [Van de Walle, 2008]:

1. Anfordern: Personen, die Ressourcen beantragen und beim Empfänger der Anfrage als vertrauenswürdig bekannt sind.
2. Beobachtung und Berichterstattung: Personen, die Beobachtungen durchführen zur besseren Einschätzung der Lage. Diese Informationen fassen sie in Berichten zusammen, die andere Beteiligte bei der Durchführung ihrer Aufgaben unterstützen.
3. Zuweisung: Personen, die bei der Zuteilung von Ressourcen die Dringlichkeit einschätzen und somit eine Priorisierung vornehmen können.
4. Lokale Aufsicht: Personen, die einen lokalen Überblick über die Situation besitzen und somit einschätzen können, ob es bei der Durchführung einer Aufgabe zu Hindernissen kommt (beispielsweise Erdbeben, Staus, überschwemmte Straßen).
5. Pflege und Wartung: Soll sicherstellen, dass eine Ressource angemessen behandelt und gewartet wird, damit sie wieder weiterverwendet werden kann.
6. Situationsanalyse und -bewusstsein: Was ist der Gesamtbedarf dieser Ressource und welche Ereignisse können auftreten, die möglicherweise die Nachfrage erhöht?
7. Globale Wiederversorgung: Zuständigkeit für das Auffinden anderer Nachschubquellen für die Erhöhung der Verfügbarkeit einer Ressource.

5.1.3 Analyse der Hauptakteure

Ein wichtiger Punkt ist das Verständnis der an einem Frühwarnsystem beteiligten Akteure, also die Identifikation der konkreten *Instanzen* der in den vorherigen Abschnitten aufgeführten Gruppierungen. Zur Modellierung der Akteure und ihrer Beziehungen eignet sich eine Ontologie. Nach Staab

ist die Spezifikation einer Ontologie auch immer ein sozialer Prozess, da sich die beteiligten Menschen auf eine Konzeptualisierung einigen müssen [Staab, 2004]. Dadurch wird das Verständnis der verschiedenen Gruppen untereinander gefördert und die Problematik aus unterschiedlichen Perspektiven beleuchtet. Hierzu eignen sich iterative Workshops, die interdisziplinär durchgeführt werden, um gemeinsam grundlegende Anwendungsfälle durchzuspielen. Teilnehmer der Workshops kommen aus den bereits identifizierten Gruppen von Akteuren. Hinzu kommen Systemarchitekten und Techniker, aber auch Soziologen und Rechtsexperten können wertvolle Beiträge leisten. Durch die formalisierte Art einer Ontologie entsteht der Zwang, die benötigten Akteure und ihre Beziehungen detailliert zu diskutieren. Dadurch steigt das Verständnis der Problematik aus verschiedenen Blickwinkeln und es wird sehr schnell deutlich, welche Aspekte noch nicht in vollem Umfang verstanden wurden. Eine erfolgreiche Anwendung dieser Vorgehensweise wird in *Modelling of an Ontology for a Communication Platform* beschrieben [Moßgraber, 2015].

5.1.4 Anwendungsfälle

Die konkreten Anwendungsfälle aus Sicht der Nutzer sind abhängig von der Domäne und müssen für jedes System spezifiziert werden. Hilfreiche Fragestellungen finden sich in Abschnitt 2.1.5. Hinweise für die Ausarbeitung der vier Phasen der Frühwarnung werden in den folgenden beiden Abschnitten gegeben.

5.1.4.1 Wissen um die Risiken und Überwachung

Zunächst müssen die zu überwachenden Indikatoren identifiziert werden. Im nächsten Schritt sind die wissenschaftlichen Grundlagen (mathematische Modelle) zur Erstellung der Prognosen zu erarbeiten. Weitere notwendige Arbeitsschritte (zum Beispiel Verifikation einer Berechnung, chemische Analysen, Interaktion mit Behörden, etc.) liegen

evtl. schon in SOPs vor. Falls dies nicht der Fall ist, müssen diese mit allen beteiligten Akteuren erstellt werden.

5.1.4.2 Kommunikation und Reaktionsfähigkeit

Für die Ausarbeitung und Dokumentation der SOPs wird BPMN empfohlen (eine vertiefende Diskussion dazu folgt in Kapitel 1). In der Entwurfsphase und zur Abstimmung mit Fachexperten eignen sich auch einfache Ablaufdiagramme, die nicht das Verständnis einer formalen Sprache voraussetzen. Abbildung 49 zeigt hierfür ein Beispiel aus dem Projekt ILEWS [Bell, 2010]. Diese Diagramme müssen aber nach der Abstimmungsphase in eine formale Beschreibung wie BPMN überführt werden. BPMN hat den Vorteil, dass die notwendigen Rollen und Akteure verwendet werden und damit ersichtlich sind. Beispiele finden sich im später folgenden Anwendungskapitel von RA-FWS^{NG}. Für die Interaktionen mit der Benutzerschnittstelle eignen sich UML Use-Cases.

5.1.5 Durchzuführende Schritte

Folgende in Tabelle 25 aufgeführten Schritte müssen zur vollständigen Ausarbeitung dieser Sicht durchgeführt werden.

Tabelle 25: Arbeitsschritte der Unternehmenssicht

#	S. Abschnitt	Arbeitsschritt
1	5.1.1, 5.1.2, 5.1.3	Analyse der Hauptakteure, der Rollen und ihrer Interaktionen.
2	5.1.4.1	Identifikation der Frühwarnindikatoren.
3	5.1.4.1	Identifikation der mathematischen Modelle und Analyseverfahren.
4	5.1.4.2	Festlegung der Anwendungsfälle mit Hilfe von SOPs.

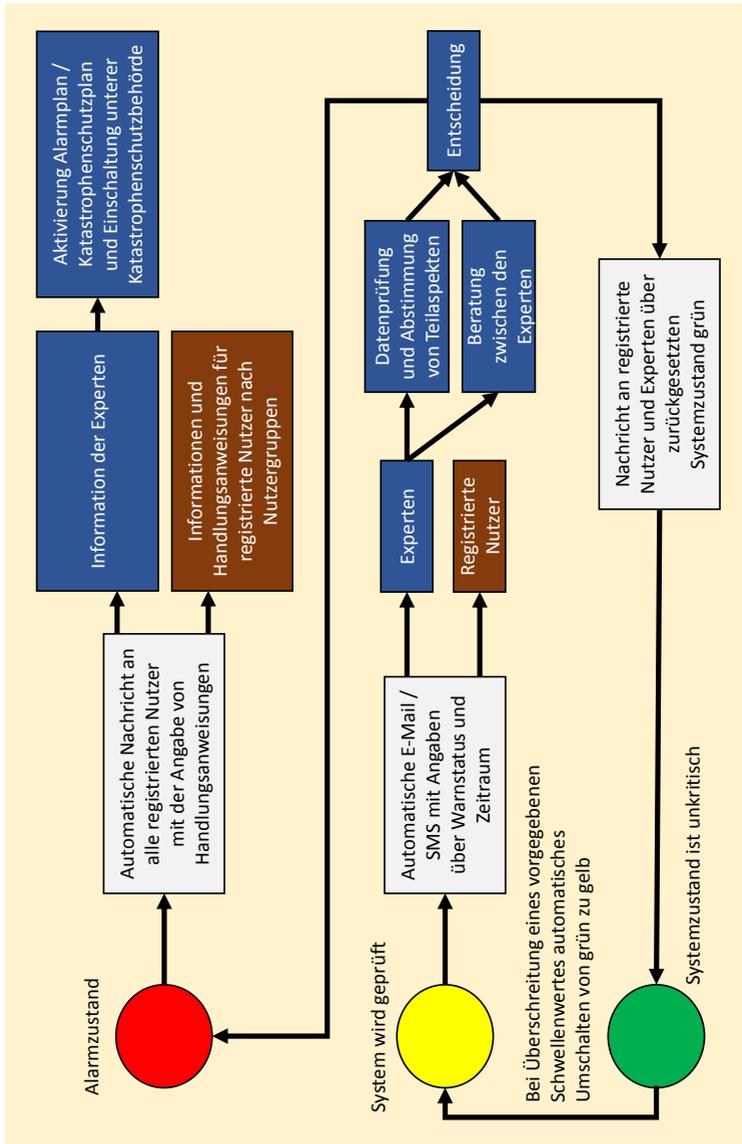


Abbildung 49: Informelle Darstellung einer SOP nach [Bell, 2010]

5.2 System von Systemen Sicht

Ein SoS ist immer mit einem erheblichen Aufwand verbunden und zwar nicht nur bei der Implementierung sondern auch beim Betrieb und der Verwaltung. Nicht immer muss ein SoS vorliegen, dies kann beispielsweise der Fall sein bei einem lokal begrenzten Frühwarnsystem, das an nur einem Standort betrieben wird. Trotzdem sollte vorgesehen werden, dieses System in ein SoS integrieren zu können.

5.2.1 Prüfung, ob ein SoS vorliegt

Die Kriterien zur Identifikation eines SoS wurden bereits in Abschnitt 2.2.3.2 beschrieben. Tabelle 26 zeigt diese Prüfung am Beispiel der Tsunami-Frühwarnung für die alle Kriterien erfüllt werden.

Tabelle 26: Prüfung auf SoS am Beispiel Tsunami-Frühwarnung

Kriterien	Tsunami-Frühwarnung (Begründung)
Operative Unabhängigkeit der Elemente	Wird von unterschiedlichen Regierungen und Institutionen betrieben, wie zum Beispiel Beobachtungszentren (National und Regional), Warnungszentren (National), Arbeitsgruppen und Regierungen (National), Wissenschaftlichen Einrichtungen (National und Regional) und Rechenzentren (National und Regional).
Unabhängigkeit der Elemente in der Verwaltung	Siehe vorheriges Kriterium.
Evolutionäre Entwicklung	Integriert fortlaufend neue Sensorik und Analysealgorithmen.
Emergentes Verhalten	Kombiniert das Wissen aus Teilsystemen.
Geographische Verteilung	Bei GITEWS beispielsweise der indische Ozean mit seinen Anrainerstaaten.

5.2.2 Die Kommunikationsschicht für ein SoS

Abbildung 50 zeigt ein SoS mit drei angeschlossenen Standorten. Die Verbindungen zwischen den Standorten müssen durch eine skalierbare und resiliente Kommunikationsschicht implementiert sein [Tao, 2012].

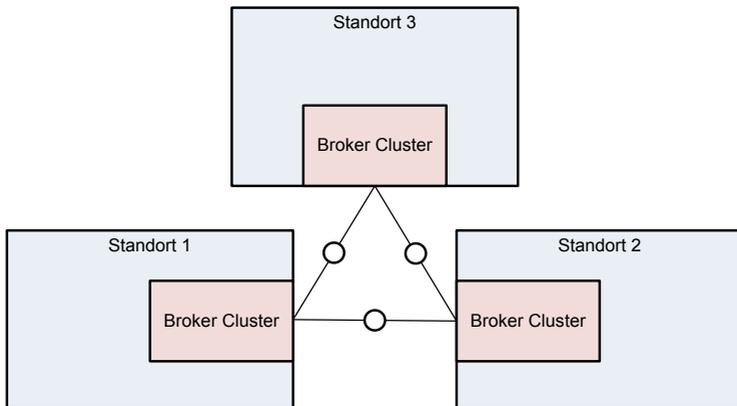


Abbildung 50: SoS mit drei angeschlossenen Standorten

5.2.2.1 Skalierbarkeit

Die Komponenten bzw. Untersysteme des Systems (s. Abschnitt 5.3) können am gleichen Standort oder an unterschiedlichen Standorten installiert sein je nach Anforderung des konkreten Frühwarnsystems. Die Komponenten können weiterhin in verschiedenen Programmiersprachen implementiert sein (Java, C#, C++, R, ...) und auf unterschiedlichen Betriebssystemen ablaufen (Linux, Windows). Diese Heterogenität verlangt nach einer standardisierten Kommunikationsschicht, die eine lose gekoppelte und asynchrone Kommunikation der heterogenen Meldungsquellen ermöglicht. Als Lösung dieser Problematik wird der Einsatz einer MOM empfohlen, sodass die Anbindung für alle Komponenten transparent ist. Die

Komponente zum Nachrichtenaustausch wird *Message Broker* genannt. An jedem Standort ist mindestens ein Broker installiert [Tao, 2012].

5.2.2.2 Resilienz

Der eingesetzte Message Broker muss die Zustellung jeder Nachricht garantieren können. Die meisten verfügbaren Softwarelösungen verwenden dazu Clustering-Techniken, d.h. mehrere Broker werden zu einem Cluster verschaltet. Bei Ausfall eines (primären) Brokers übernimmt ein anderer (gespiegelter) Broker dessen Aufgabe. Jedoch stellt dieser Entwurf keine Garantie für einen Verbindungsausfall dar und benötigt mehr Speicherplatz für die Backups der Nachrichten. Um diese Einschränkungen zu überwinden, können u.a. folgende Techniken eingesetzt werden [Tao, 2012].

Spiegelung von Topics

Topics werden in einem Broker verwendet, um Nachrichten zu einem Thema zu gruppieren. Topic-Spiegelung wird verwendet, um Nachrichten bestimmter Topics sowohl im primären Broker als auch im gespiegelten Broker zu replizieren, d.h. wenn einer der Broker ausfällt, können diese Nachrichten noch weiterverbreitet werden.

Self-healing

Selbstheilung (engl. Self-healing) ist die Idee, dass das System selbständig den Status, der vor dem Ausfall einer Komponente vorlag, wiederherstellen kann [Richling, 2011]. Fällt beispielsweise ein Broker aus, wird dieser automatisch neu gestartet. Während der Ausfallzeit wird den verbundenen Klienten transparent ein neuer Broker zugeteilt.

Redundantes WAN-Routing

Für den Nachrichtenaustausch über ein WAN (engl. Wide Area Network) von einem zu einem anderen Broker sollten alternative Pfade verfügbar sein (engl. Overlay-paths). Der zu verwendende Weg wird dann auf der

Grundlage des aktuellen Verbindungsstatus ausgewählt. Darüber hinaus wird eine redundante Verbindung eingeführt. Die Nachrichten werden dadurch über mehrere Wege an das Ziel geleitet, sodass selbst beim Abbruch einer Verbindung die Nachrichten noch durch den redundanten Pfad zugestellt werden können.

5.2.3 Durchzuführende Schritte

Folgende in Tabelle 27 aufgeführten Schritte müssen zur vollständigen Ausarbeitung dieser Sicht für ein konkretes Frühwarnsystem durchgeführt werden.

Tabelle 27: Arbeitsschritte der Sicht System von Systemen

#	S. Abschnitt	Arbeitsschritt
1	5.2.1	Prüfung, ob ein SoS vorliegt.
2 (opt.)	5.2.2	Ggf. Implementierung einer skalierbaren und resilienten Kommunikationsschicht.

5.3 Systemsicht

Die Systemsicht hat die Zerlegung des Systems in Komponenten, die über Schnittstellen interagieren, im Fokus. Weiterhin beschreibt sie die im System ablaufenden Prozesse, Berechnungen und Algorithmen. Die Komponenten des Systems (Dienste, Module, etc.), ihre Schnittstellen und ihr Zusammenwirken werden dokumentiert. Für die in diesem Kapitel gezeigten Diagramme wurde FMC verwendet. Abbildung 51 zeigt die Systemsicht von RA-FWS^{NG}, deren Komponenten in den folgenden Abschnitten näher beschrieben werden. Die Darstellung enthält die Komponenten, die an einem Standort vorhanden sind. Einzelne Standorte können, wie in der SoS-Sicht beschrieben, verbunden sein.

5.3.1 Message-oriented Middleware

Die MOM wurde bereits in der SoS-Sicht thematisiert (s. Abschnitt 5.2.2).

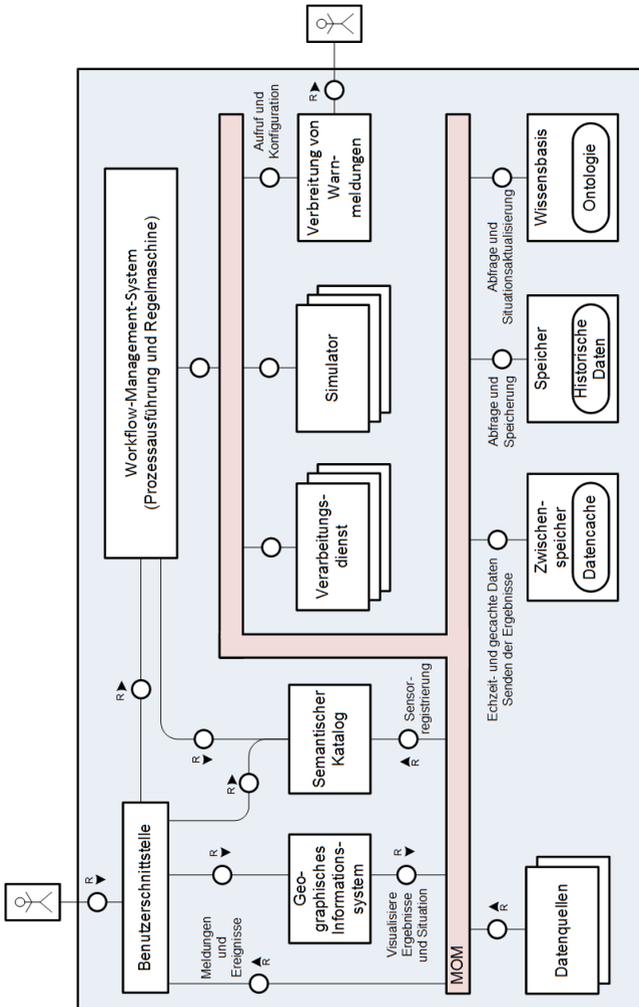


Abbildung 51: Systemsicht RA-FWS^{NG}

5.3.2 Datenquellen

Datenquellen veröffentlichen ihre Daten über die MOM (einen Broker). Bevor sie sich als Datenquelle in das System einbringen können, müssen sie sich beim semantischen Katalog anmelden und Informationen über ihre Sensorik, Datentypen und -formate zur Verfügung stellen. Eine Datenquelle kann ein einzelner Sensor, ein komplettes Sensornetzwerk oder sogar ein ganzes System sein. Im Industrieumfeld oder bei Steuerungsanlagen für kritische Infrastrukturen (zum Beispiel Wasser- oder Atomkraftwerk) trifft man häufig auch auf SCADA Systeme (s. Technologiesicht), die als Datenquelle dienen. Die Ausgabe einer Datenquelle sollte im Optimalfall nicht nur aus Rohdaten bestehen, sondern bereits semantische Annotationen enthalten, um diese leichter in die Wissensbasis integrieren zu können. Dies hat weiterhin den Vorteil, dass die Ereignisnachrichten, die über die MOM gesendet werden, „eine gemeinsame Sprache sprechen“. Schritte für die Datenintegration werden in der Informationssicht beschrieben.

5.3.3 Zwischenspeicher

Einige Auswertelgorithmen und Simulatoren benötigen keine Einzelwerte, die in Echtzeit eingespeist werden, sondern gesammelte oder bereits aggregierte Daten über einen Zeitraum (zum Beispiel eine Stunde).

Die Zwischenspeicherkomponente abonniert sich auf die jeweils gewünschten Datenkanäle der MOM und schreibt die empfangenen Daten in den Datenbankcache. Der Cache kann dann von den jeweiligen Verarbeitungsdiensten und Simulatoren zugegriffen werden, die das Caching bei der Zwischenspeicherkomponente angefordert haben.

5.3.4 Speicher

Alle Meldungen, die über die MOM gesendet werden, können im Speicher für spätere oder langlaufende Auswertungen abgelegt werden. Um die

Größe des Speichers klein halten zu können, muss er die Möglichkeit unterstützen, diverse positive oder negative Filterkriterien zu setzen. Abhängig von der Größe der zu speichernden Daten des Standorts muss das Datenarchiv fähig sein, auch mit Massendaten umzugehen.

Die Schnittstelle zur Datenabfrage bietet Zugriff auf unterschiedliche Daten und -typen, die sich entsprechend ändern können, wenn neue (Sensor-) Komponenten dem System hinzugefügt werden. Deshalb müssen Abfragen nach den verfügbaren Datentypen und ihren Eigenschaften möglich sein.

5.3.5 Wissensbasis

Die Wissensbasis speichert die mittels Verarbeitungsdiensten gewonnen Erkenntnisse in einer Ontologie. Der Inhalt der Ontologie ist in der Informationssicht beschrieben. Auf die Konstruktion der Wissensbasis und die Zugriffsmöglichkeit wird in Abschnitt 6.4 näher eingegangen.

5.3.6 Verarbeitungsdienste

In einem System, das aus einer flexiblen Anordnung von Sensoren und Sensoranalyse-Software besteht, können verausachte, redundante und widersprüchliche Messungen und Beobachtungen auftreten. Wenn hierfür keine Vorkehrungen getroffen werden, verursachen diese Bedingungen zwangsläufig eine fragmentierte und doppelte Darstellung eines Objekts aus der realen Welt in der Datenbank. Wird diese Unschärfe jedoch korrekt modelliert, kann die Qualität der Informationen in der Datenbank erheblich durch die Aggregation redundanter Informationen verbessert werden und so Rauschen und Falscherkennungen reduzieren.

5.3.6.1 Datenkonsolidierung und -fusion

Das Ziel der Konsolidierung und Fusion ist es daher Beobachtungen, die sich auf das gleiche Objekt beziehen, zusammenzufassen. Dies geschieht durch Ausnutzung

- redundanter Informationen in den Beobachtungen
- mathematischer Modelle der Objekte aus der realen Welt (und ihr Verhalten)
- von Modellen über mögliches Beobachtungsrauschen und die Falschalarmrate

Da die Modelle und die Strukturen der Information, die fusioniert werden sollen, nicht zur Entwurfszeit festgelegt werden können, ist eine Plug-In-Architektur vorgesehen. Neue Modelle und Algorithmen für die Fusion können über den semantischen Katalog hinzugefügt werden.

5.3.6.2 Auswertelgorithmen

Algorithmische Prozesse arbeiten auf Daten, die von Datenquellen oder anderen algorithmischen Prozessen veröffentlicht werden und die deshalb auch als Datenquellen angesehen werden können. Die Daten können je nach Anforderung des Algorithmus entweder direkt über die MOM, vom Datenbankarchiv oder durch den Zugriff auf den Zwischenspeicher empfangen bzw. gelesen werden. Existierende Fusionsalgorithmen können in folgende Kategorien eingeordnet werden [Klopfer, 2009]:

- Räumliche Fusion: In-Situ Sensorik ist nicht flächendeckend vorhanden. Werte für die nicht abgedeckten Bereiche können aus den umliegenden Sensoren interpoliert werden. Beispiele für Algorithmen sind der Kriging-Algorithmus und die Bayessche Maximum-Entropie-Methode [Christakos, 2001].
- Kausale Fusion: Statistische Analyseverfahren, die zum Ziel haben, Beziehungen zwischen einer abhängigen und einer oder mehreren unabhängigen Variablen zu modellieren. Beispiele sind lineare Regression oder neuronale Netze [Christakos, 2001].
- Zeitliche Fusion: Bei der zeitlichen Fusion werden historische Werte einer Variable verwendet, um die Änderung des aktuellen Wertes vorherzusagen, beispielsweise mit einem Kalman-Filter [Christakos, 2001].

5.3.6.3 Interaktion der Verarbeitungsdienste

Die Integration der Verarbeitungsdienste in die Systemarchitektur und den Verarbeitungsablauf zeigt Abbildung 52.

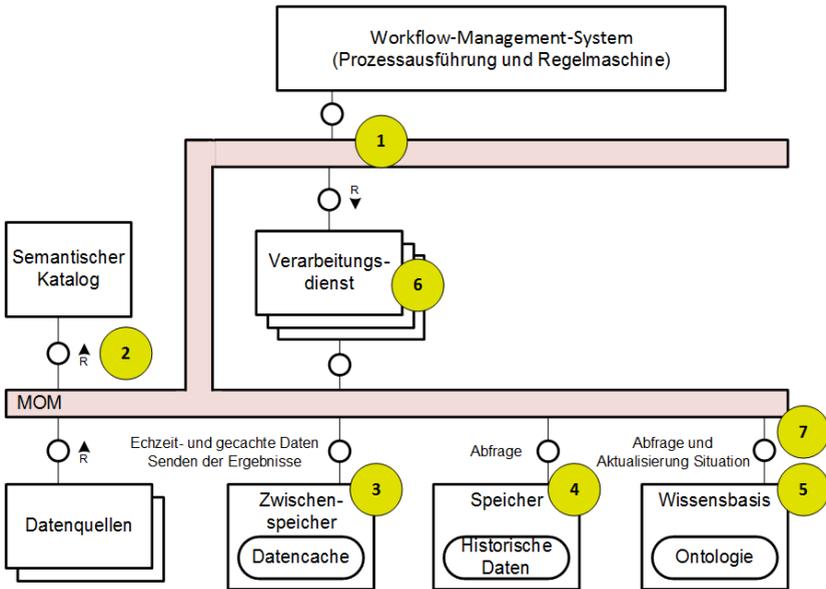


Abbildung 52: Ablauf und Interaktion von Verarbeitungsdiensten

1. Für den Arbeitsbeginn eines Verarbeitungsdienstes gibt es die folgenden drei Fälle.
 - a. Ein Dienst kann dauerhaft aktiv sein und nimmt seine Arbeit direkt beim Systemstart auf. Dies sind meist Dienste, die Basisdaten berechnen, wie beispielsweise das aktuelle Strömungsverhalten in einem Wassernetzwerk.
 - b. Auch in diesem Fall führt der Dienst fortlaufende Berechnungen aus, wird aber während der Laufzeit vom Workflow-Management-System auf andere Parameter konfiguriert.
 - c. Ein Dienst erhält einen Auftrag zu einer Berechnung und beendet sich wieder nach dessen Abarbeitung. Der Auftrag wird über die MOM erteilt und kann vom Workflow-Management-System oder einem auftretenden Ereignis (entweder von der Sensorik oder anderen Verarbeitungsdiensten) ausgelöst werden.
2. Der Dienst kann den semantischen Katalog verwenden, um nach Datenquellen für einen bestimmten Messwert im gewünschten räumlichen und zeitlichen Bereich zu suchen.
3. Der Verarbeitungsdienst liest die gewünschten Daten aus den gefunden Datenquellen und abonniert sich dazu auf Echtzeitdaten. Bestimmte Dienste benötigen Zeitscheiben statt Einzelwerte, die vom zwischengeschalteten Cache geliefert werden.
4. Alternativ kann der Dienst auch historische Daten für seine Berechnung einsetzen.
5. Dienste auf höherer Ebene (beispielsweise zur Situationsanalyse) arbeiten nicht direkt auf den Rohdaten, sondern auf den semantischen Informationen der Wissensbasis.
6. Der Dienst führt seine Berechnungen aus.
7. Die Ergebnisse der Berechnungen werden in der Wissensbasis gespeichert.

5.3.6.4 Zugriff auf große Datenmengen durch Algorithmen

Zunächst ist zu klären, ob tatsächlich ein BigData Problem vorliegt. Ähnlich wie bei SoS ist die Integration und Verwaltung einer Verarbeitungsumgebung für BigData mit einem erheblichen Aufwand verbunden. Zur Einschätzung der Notwendigkeit muss eine Hochrechnung des Datenaufkommens der eingesetzten Sensorik durchgeführt werden, die auch den zukünftigen Ausbau des Systems soweit absehbar beinhaltet. Liegt ein BigData-Problem vor, sollte geprüft werden, ob eine Umgehung der Problematik möglich ist. Zur Reduzierung der Datenmenge können beispielsweise Sensordaten direkt am Sensor aggregiert oder nur in bestimmten Zeitintervallen (zum Beispiel stündlich), oder wenn ein Grenzwert überschritten wird, gesendet werden. Ist eine BigData Verarbeitung notwendig, kann die Lambda-Architektur (s. Abschnitt 3.1.5) in RA-FWS^{NG} integriert werden. Die Verarbeitungskette der Lambda-Architektur wird dabei als einzelner Verarbeitungsdienst betrachtet und ist in der gleichen Weise, wie oben beschrieben, integriert.

Problematischer wird es, wenn die Daten an einem anderen Standort (Standort A) gespeichert sind als auf dem, in dem der Algorithmus läuft (Standort B) und keine standortübergreifende Lambda-Architektur umgesetzt werden kann, da es unterschiedliche Zuständigkeiten gibt. In diesem Fall sind folgende drei Lösungsmöglichkeiten denkbar:

1. Der Algorithmus fordert die Daten an und bekommt ein großes Datenpaket, welches die archivierten Daten beinhaltet. Die archivierten Daten müssen in einem komprimierten Format zugänglich gemacht werden, die auf die Client- bzw. Algorithmus-Seite effizient übertragen und verarbeitet werden kann. Eine technische Lösung hierfür ist, den Antrag über die MOM auszulösen, einen Download-Link als Antwort zu erhalten und dann auf die Daten über einen FTP-Server zuzugreifen.
2. Der gewünschte Algorithmus ist auch auf Standort A verfügbar, was bedeutet, dass die Verarbeitung auf dem gleichen Standort

durchgeführt werden kann, an dem auch die Daten gespeichert sind und somit nur die Ergebnisse zurück an den Standort B übertragen werden müssen. Die Übertragung der (aggregierten und damit kleinen) Ergebnisse kann in diesem Fall wiederum über die MOM erfolgen.

3. Der Algorithmus wird von Standort B auf den Standort A übertragen und dort ausgeführt. Diese Lösung ist zwar die flexibelste, hat aber mehrere Nachteile. Der Algorithmus muss in einer Art und Weise geschrieben und verpackt werden, sodass er auf Standort A ausgeführt werden kann (was evtl. mit der Portierung des Codes in eine andere Programmiersprache und Umgebung verbunden ist). Darüber hinaus sind viele Sicherheitsprobleme in Betracht zu ziehen.

5.3.7 Simulationen

Für Simulationen gelten die gleichen Aussagen, die für Auswertelgorithmen gemacht wurden. Bei Simulationen ist jedoch zu beachten, dass eine stärkere Interaktion mit dem Anwender notwendig ist, um beispielsweise alternative Szenarien testen zu können. Die Kommunikation erfolgt dabei über das Workflow-Management-System. Simulationen sollten für folgende Kategorien vorgesehen werden [Bernard, 2015b]:

- Echtzeitanalysen: Wie ist die aktuelle Situation? Meist werden hier auch Basisdaten berechnet, die als Eingabe für andere Simulationen benötigt werden, zum Beispiel das aktuelle Strömungsverhalten in einem Trinkwassernetz oder die Wettersituation mit Windverhältnissen.
- Zukunftsprognosen: Wie verändert sich die Situation? Beispiele sind Ausbreitungsrechnungen (Radioaktivität, Verunreinigungen) zu denen auch die sogenannten „What-if“-Analysen gehören: Wie verändert sich die Ausbreitung, wenn ich eine bestimmte Maßnahme treffe?

- Rückrechnungen zur Ursachenanalyse: Was ist die Ursache der Situation? Zum Beispiel, an welcher Stelle kam die Verunreinigung ins Trinkwassernetz?

5.3.8 Verbreitung von Warnmeldungen

In den verschiedenen Schritten eines Arbeitsablaufs zur Behandlung einer Gefahr müssen die Öffentlichkeit bzw. nachgelagerte Zuständigkeitsinstanzen, wie zum Beispiel Polizei, Feuerwehr und Katastrophenhelfer informiert werden. Die Komponente zur „Verbreitung von Warnmeldungen“ kümmert sich um die Informationslogistik und ist für die Erzeugung von empfängerspezifischen Warnungen verantwortlich. Abbildung 53 zeigt die Detaildarstellung der Komponente und ihre Interaktionen.

1. Die Komponente erhält den Auftrag zur Verbreitung einer Meldung vom Workflow-Management-System.
2. Der Inhaltsgenerator selektiert die notwendigen Informationen aus der Wissensbasis.
3. Das Benachrichtigungssystem wählt die konkreten zu informierenden Personen der jeweiligen Organisation aus. Die Personen ergeben sich aus dem hinterlegten Notfallplan der jeweiligen Organisation, in welchem die Entscheider dokumentiert sind. Für jeden Empfänger muss die richtige Sprache, Formulierung und der passende technische Kanal gewählt werden. Diese Personalisierungen für einen Empfänger bzw. eine Empfängergruppe sind in Profilen in der Wissensbasis hinterlegt. Das Benachrichtigungssystem liest diese Profile und ruft für jeden Adressaten den Nachrichtengenerator und die Zustellungskomponente auf.
4. Der Nachrichtengenerator erzeugt auf Basis der erhaltenen Meldungsdaten und des Empfängerprofils die Nachricht.

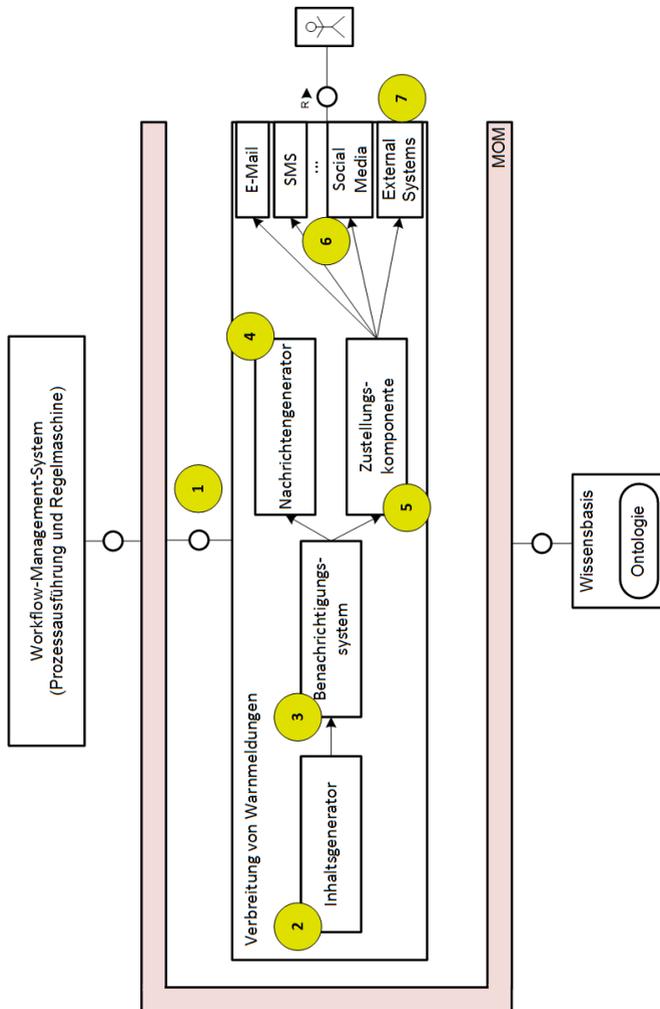


Abbildung 53: Aufbau und Interaktion der Komponente zur Verbreitung von Warnmeldungen

5. Die Zustellungskomponente liefert die Nachricht über den gewünschten Kanal (oder auch mehrere), wie zum Beispiel E-Mail, SMS oder soziale Medien aus. Die Warnmeldung kann je nach

Zuständigkeit des Meldungsherausgebers auch in ein System einer nachgelagerten Instanz (zum Beispiel Katastrophendienst) übertragen werden, die dann ihrerseits die korrekte Sprache und Formulierung anwendet.

5.3.9 Workflow-Management-System

In aktuellen Frühwarnsystemen sind die Informationslogistikprozesse noch weitestgehend fest verdrahtet. Dies erschwert die Anpassung der Abläufe, falls neue Erfahrungen (zum Beispiel durch Übungen) oder rechtliche Vorschriften vorliegen. Daher wird ein BPMS (hier als *Workflow-Management-System* bezeichnet) verwendet, um Frühwarnprozesse und Krisenmanagementpläne zu automatisieren bzw. den Benutzer zu führen. Workflows neigen dazu, schnell unübersichtlich zu werden, wenn komplexe Entscheidungskriterien modelliert werden müssen [Rücker, 2016]. Dies ist gerade bei Frühwarnsystemen regelmäßig der Fall. Als Lösung dieses Problems empfiehlt sich die Integration von Entscheidungstabellen, um regelbasierte adaptive Workflows zu realisieren. Entscheidungstabellen sind ein erprobtes Werkzeug und werden zum Beispiel intensiv im Finanzwesen eingesetzt, um komplexe Regelsätze durch einen Fachexperten modellieren zu lassen. Im Workflow taucht dann nur ein verständlich benannter Entscheidungspunkt auf, der intern von Regelinterpreter (engl. Rule-Engine) abgearbeitet wird. Eine nähere Diskussion dieser Komponente und die Integration von Regeln erfolgt in Kapitel 1.

5.3.10 Semantischer Katalog

Der *Semantische Katalog* (engl. Semantic Registry) ermöglicht den systemweiten Zugriff auf die Positionen und Eigenschaften der verfügbaren Sensorik in standardisierter Form. Dies ermöglicht eine flexible Erweiterung des Systems um neue Sensorik bzw. den Austausch existierender Sensorik. Nutzer des Katalogs können zum Beispiel die

verfügbare Sensorik mit bestimmten gewünschten Eigenschaften in einem geographischen Gebiet abfragen.

Der Semantische Katalog unterstützt die Verwaltung statischer und dynamischer Eigenschafts- und Konfigurationsdaten von Sensoren. Als Austauschformat der Daten wird jeweils eine Instanz eines spezifischen Sensorprofils des Sensortyps benötigt. Sensoren mit gleicher Ausgabe teilen sich das gleiche Sensorprofil, zum Beispiel alle Sensoren zur Messung seismischer Aktivität. Alle Sensorprofile enthalten jedoch einen Raumbezug. Dies sind zum einen ihre geographische Position sowie der von ihr abgedeckte Bereich. Dies ermöglicht die geographische Abfrage und zum Beispiel die Darstellung auf einer Karte.

Neben der Abfrage statischer und dynamischer Fähigkeiten unterstützt die Schnittstelle zum semantischen Katalog auch das dynamische Hinzufügen und Entfernen von Sensoren. Eine nähere Diskussion dieser Komponente erfolgt in Abschnitt 6.2.

5.3.11 Geographisches Informationssystem

Die GIS Komponente visualisiert mehrere geographische Informationen. Dazu greift die Komponente auf mehrere Datenquellen zu (s. Abbildung 54).

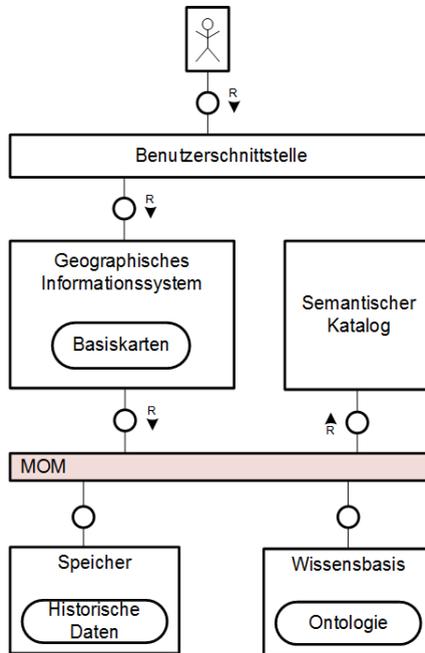


Abbildung 54: Integration der GIS Komponente in RA-FWS^{NG}

Unterschiedliche Informationen werden in Ebenen (engl. Layer) gegliedert, die ein- und ausgeblendet werden können. Informationsebenen (Punkt 2 bis 3) werden über Basiskarten gelegt.

1. Basiskarten: Diese zeigen Informationen über das gewünschte Gebiet an wie Flüsse, Seen, Straßen, Gebäude oder Höhenmodelle. Weitere Basiskarten mit Bezug auf die Domänen können vorhanden sein, beispielsweise Netzwerkkarten bei kritischer Infrastruktur.
2. Speicher: Rohdaten oder vorverarbeitete Daten, zum Beispiel Ozonmesswerte, die farblich nach ihrer Intensität über die Karte interpoliert werden.

3. Wissensbasis: Berechnete Simulationsergebnisse (Ausbreitung eines Gefahrstoffes) oder spezifische Ereignisse, die an ihrer geographischen Position als Markierung angezeigt werden wie etwa das Epizentrum eines Erdbebens oder die Position von Rettungskräften.
4. Semantischer Katalog: Die im Katalog angemeldete Sensorik kann als Kartenebene angezeigt werden. Zusätzlich ist auch die geographische Suche nach Sensorik über das GIS möglich, beispielsweise kann der Nutzer ein Rechteck aufziehen, um die darin vorhandene Sensorik anzuzeigen.

5.3.12 Benutzerschnittstelle

Die Benutzerschnittstelle integriert alle Funktionalitäten, die das System zur Verfügung stellt, in eine einheitliche Oberfläche. Dadurch wird sichergestellt, dass kein Zeitverlust durch manuellen Datenaustausch mit zusätzlichen Informationssystemen (zum Beispiel einem separaten Simulationssystem) entsteht. Jedoch gibt es unterschiedliche Zugänge zum System, die auf bestimmte Einsatzzwecke zugeschnitten sind. Akteure in einer Einsatzzentrale benötigen einen Arbeitsplatz, auf dem die Lage dargestellt werden kann (zum Beispiel ein Desktopsystem mit Projektion auf eine Leinwand). Akteure im Einsatz benötigen dagegen mobile Geräte wie Smartphones oder Tablets. Die Oberfläche passt sich automatisch mit ihren Darstellungsformen an das jeweilige Endgerät an. Die Benutzeroberfläche unterstützt den Benutzer des Systems bei der Entscheidungsfindung mit Hilfe von:

- Anzeigen von Daten in Diagrammen
- Anzeigen von Daten in Tabellen
- Anzeigen von Daten auf Karten
- Anzeige von Warnungen und Ereignissen
- Verwaltung von Simulationen und die Visualisierung ihrer Ergebnisse

- Integration von Dokumenten und weiteren Datenquellen für Recherchen der Domäne

Die Benutzerschnittstelle ist jeweils sehr spezifisch für ein konkretes System und muss deshalb in Abstimmung mit den Akteuren spezifiziert werden.

5.3.13 Durchzuführende Schritte

Folgende in Tabelle 28 aufgeführten Schritte müssen zur vollständigen Ausarbeitung dieser Sicht durchgeführt werden.

Tabelle 28: Arbeitsschritte der Systemsicht

#	S. Abschnitt	Arbeitsschritt
1	5.3.9	Umsetzung der in der Unternehmenssicht definierten SOPs.
2	5.3.2	Integration aller notwendigen Datenquellen.
3	5.3.6	Anbindung und Implementierung aller Verarbeitungsdienste.
4	5.3.6.4	Prüfung, ob ein BigData Problem vorliegt.
5 (opt.)	5.3.6.4	Integration BigData Verarbeitung.
6	5.3.7	Anbindung und Implementierung aller Simulationen.
7	5.3.11	Identifikation und Integration des benötigten Kartenmaterials.
8	5.3.12	Spezifikation der Benutzeroberfläche.

5.4 Informationssicht

Die Informationssicht klassifiziert und spezifiziert die Informationsanforderungen in Bezug auf ein Informationsmodell. Dieses beinhaltet die Informationsobjekte, ihre Attribute und die Beziehungen untereinander.

5.4.1 Modellierung der Sensorik

Zur Beschreibung von Sensoren wird SensorML eingesetzt. SensorML beschreibt ein Informationsmodell, welches das Auffinden und Nutzbar-machen von verfügbaren Sensoren unterstützt. Innerhalb dieses Kontextes erlaubt SensorML dem Entwickler, Modelle und XML-Schemata zu definieren sowie jeglichen Prozess inklusive der Messungen eines Sensorsystems zu beschreiben und zu verarbeiten. Der letztere Aspekt bedeutet, dass ein Sensor auch ein komplexes System von anderen Sensoren mit Prozessen sein kann. Beispielsweise könnte der Fahrweg eines Hilfskonvois als ein Sensor beschrieben werden, dessen Messwerte aus Satellitenbildern mit einem Bildanalyseprozess erzeugt werden. Für die spezifischen, in einem System verfügbaren Sensortypen muss jeweils eine SensorML Beschreibung erstellt werden.

5.4.2 Modellierung von Messwerten

Für Messwerte wird O&M eingesetzt. Die O&M-Spezifikation enthält allgemeine Modelle für Sensorbeobachtungen und Messungen. Sie wird in allen anderen SWE Standards als Datenmodell verwendet. Die folgenden Definitionen aus dem O&M-Standard sind von besonderem Interesse (Abbildung 55 zeigt dazu ein Beispiel).

- Beobachtung (engl. Observation): Eine Beobachtung ist eine Aktion, die mit einem diskreten Zeitpunkt oder Zeitraum verbunden ist und durch die eine Zahl, ein Begriff oder ein anderes Symbol einem Phänomen zugeordnet wird.
- Ergebnis (engl. Result): Das Ergebnis enthält den Wert, der durch das (Mess-)Verfahren erzeugt wird. Die Art des Beobachtungsergebnisses muss mit der beobachteten Eigenschaft und dem Wertebereich ihrer Einheit übereinstimmen.
- Prozess (engl. Procedure): Das (Mess-)Verfahren, das verwendet wird, um das Ergebnis zu erzeugen.

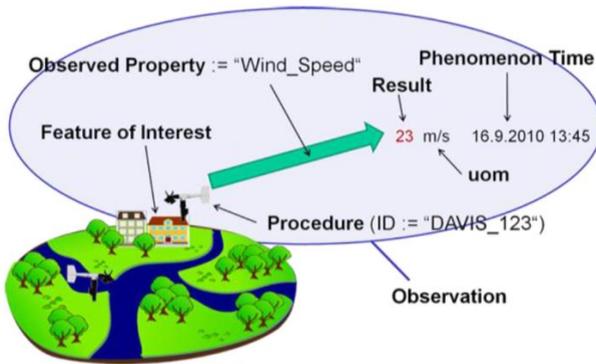


Abbildung 55: Bestandteile einer O&M Beobachtung [Bröring, 2011]

- Merkmal von Interesse (engl. Feature of Interest): Das Merkmal von Interesse ist ein Objekt der realen Welt, an dem eine Beobachtung vorgenommen wird.
- Beobachtete Eigenschaft (engl. Observed Property): Identifiziert oder beschreibt das Phänomen, für die das Beobachtungsergebnis eine Schätzung seines Wertes zur Verfügung stellt. Es muss eine Eigenschaft mit dem Typ des Feature of Interest in Verbindung gebracht werden.
- Einheit (engl. Unit of Measure): Ermöglicht die Interpretation des Ergebniswerts. Meist werden die Codes von Unified Code for Units of Measure (UCUM)¹ verwendet.

5.4.3 Inhalt der MOM Nachrichten

Als Format der Meldungen, die über die MOM ausgetauscht werden, wird ebenfalls das O&M-Modell verwendet. Aus verschiedenen Gründen, wie beispielsweise die Übertragungsgeschwindigkeit oder die notwendige

¹ <http://unitsofmeasure.org/trac>, letzter Abruf am 4.9.2016

Unterstützung bestehender Standards einer Domäne, können auch andere Nachrichtentypen unterstützt werden.

5.4.4 Ontologie für die Wissensbasis

Für O&M sind bereits mehrere Übersetzungen in eine Ontologie verfügbar, beispielsweise SSNO [Compton, 2012]. Die SSNO ist komplex, deshalb wird empfohlen, diejenigen Konzepte und Eigenschaften zu extrahieren, die auch eingesetzt werden. Als Basis können die folgenden Konzepte und Eigenschaften verwendet werden [Moßgraber, 2014]:

- Observation
 - hasObservedProperty
 - measuredByProcedure
 - hasValue
 - relatesToFeatureOfInterest
 - hasTime
 - hasUnit
- Phenomenon
 - hasName
 - hasID
- Procedure
 - hasName
 - hasID
- FeatureOfInterest
 - hasName
 - hasID
 - hasNorthing
 - hasEasting

Die Integration weiterer Ontologien zur Modellierung von u.a. Zeit, Raum, Ereignissen und Domänenwissen ist als Arbeitsschritt durchzuführen.

5.4.5 Durchzuführende Schritte

Folgende in Tabelle 29 aufgeführten Schritte müssen zur vollständigen Ausarbeitung dieser Sicht für ein konkretes Frühwarnsystem durchgeführt werden.

Tabelle 29: Arbeitsschritte der Informationssicht

#	S. Abschnitt	Arbeitsschritt
1	5.4.1	Erstellung der SensorML Beschreibungen für die spezifischen Sensortypen.
2	5.4.4 / 6.1	Integration von Domänen-Ontologien.

5.5 Technologiesicht

Die Technologiesicht legt die Technologien fest, mit der das System implementiert wird bzw. arbeitet. Insbesondere wird auch auf verwendete Standards eingegangen. Die verfügbare Hard- und Software entwickelt sich in sehr kurzen Zyklen weiter. Deshalb kann in den folgenden Abschnitten nur ein Überblick der aktuell verfügbaren Technologien gegeben werden. Für die aktuelle Erstellung eines Frühwarnsystems muss deshalb immer eine Analyse des aktuellen Standes der Technik vorgeschaltet werden.

5.5.1 Message-oriented Middleware

Bei Technologien zur Umsetzung einer MOM ist zu unterscheiden zwischen der eigentlichen Brokersoftware, den von ihr unterstützten Kommunikationsprotokollen und den verfügbaren Programmierschnittstellen zum Versenden und Abrufen der Nachrichten.

5.5.1.1 Protokolle

*Message Queue Telemetry Transport (MQTT)*² ist ein offenes Nachrichten-Protokoll, das von der ISO (ISO/IEC 20922:2016) und OASIS standardisiert wurde [Banks, 2014]. Es wird hauptsächlich für die M2M eingesetzt und ermöglicht die Übertragung von Telemetrie-Daten in Form von Nachrichten zwischen Geräten trotz hoher Verzögerungen oder beschränkten Netzwerken. Entsprechende Geräte reichen von Sensoren und Aktoren, Mobiltelefonen, eingebetteten Systemen in Fahrzeugen oder Laptops bis zu voll entwickelten Rechnern. MQTT hat im Bereich von IoT starke Verbreitung gewonnen und ist in RA-FWS^{NG} interessant für die Anbindung der Sensorik.

Das *Advanced Message Queuing Protocol (AMQP)* ist ein offener OASIS und ISO Standard [Godfrey, 2012] und stellt ein binäres Netzwerkprotokoll dar. AMQP ist unabhängig von der Programmiersprache. Um der großen Verbreitung von JMS (s.u.) Rechnung zu tragen, sind alle Funktionen von JMS in das Protokoll eingearbeitet worden. Dies ermöglicht es den Entwicklern, weiter die JMS-Schnittstelle zu nutzen, während sich MOMs untereinander mit AMQP verständigen können.

5.5.1.2 Programmierschnittstellen

Der *Java Message Service (JMS)* ist eine Programmierschnittstelle (API) für das Senden und Empfangen von Nachrichten über eine MOM [Hapner, 1999]. Die API ist spezifisch für die Programmiersprache Java.

5.5.1.3 Broker

*Apache ActiveMQ*³ ist ein freier Message Broker, der in Java programmiert ist und vollständig den JMS Standard implementiert. Abgesehen von JMS unterstützt er die Protokolle AMQP, STOMP (The Simple Text Oriented

² <http://mqtt.org/>, letzter Abruf am 4.9.2016

³ <http://activemq.apache.org/>, letzter Abruf am 4.9.2016

Messaging Protocol)⁴ und das ActiveMQ eigene Protokoll OpenWire. Neben Java werden auch die Programmiersprachen C#, C, Ruby, Perl, Python und PHP unterstützt. *Apache Qpid* ist ein weiterer freier Message Broker, der als Protokoll AMQP verwendet. Als Programmiersprachen werden C, C++, Java, und Python unterstützt. Weiterhin wird JMS unterstützt. Beispiele für MQTT Broker sind die kommerzielle Software *HiveMQ*⁵, die von der deutschen Firma dc-square GmbH entwickelt wird, und die offene Software *Mosquitto*⁶. Beide unterstützen das vollständige MQTT Protokoll.

5.5.2 Datenquellen

In Tabelle 30 sind relevante Standards für den Zugriff auf Sensordaten zusammengefasst. Neben den bereits bekannten OGC Standards sind Standards für den Zugriff auf SCADA Systeme aufgeführt. Diese sind besonders relevant für Frühwarnsysteme, die Daten aus industriellen Anlagen integrieren müssen, wie beispielsweise bei Atomkraftwerken oder in der Trinkwasserversorgung.

Tabelle 30: Standards zum Zugriff auf Sensordaten

<p>Sensor Observation Service</p>	<p>S. Abschnitt 2.2.5.2.</p> <p>Die Besonderheit des Standards ist, dass jegliche Art von Sensoren, von einfachen, fest installierten Messeinheiten (zum Beispiel Thermometer) über mobile Sensoren bis hin zu bilderzeugenden Fernerkundungssystemen (zum Beispiel Satelliten), durch den SOS verwaltet werden können. Der Sensor Observation Service besitzt drei obligatorische Kern-Operationen:</p> <ul style="list-style-type: none"> • GetCapabilities: Liefert Service-Metadaten zum Sensor Observation Service. • DescribeSensor: Liefert detaillierte Informationen über einen Sensor und dessen Prozesse, welche die
-----------------------------------	---

⁴ <https://stomp.github.io/>, letzter Abruf am 4.9.2016

⁵ <http://www.hivemq.com/>, letzter Abruf am 4.9.2016

⁶ <https://mosquitto.org/>, letzter Abruf am 4.9.2016

	<p>Messungen generieren. Die Beschreibung erfolgt in SensorML.</p> <ul style="list-style-type: none"> • GetObservation: Ermöglicht den Zugang zu Beobachtungen und Messungen von Sensoren, die zusätzlich gefiltert werden können. Die Beschreibung der Messdaten und deren Austausch werden mit Hilfe der SWE-Standardsprache O&M realisiert. <p>Neben weiteren zusätzlichen optionalen Operationen sind zwei transaktionale Operationen, RegisterSensor (Anlegen eines neuen Sensors) und InsertObservation (Einfügen von neuen Beobachtungen), im SOS verfügbar. Durch den SOS ist eine interoperable webbasierte Möglichkeit zum Auffinden, Einbinden und Abfragen von Sensoren und deren Messwerten gegeben.</p>
SensorThings API	S. Abschnitt 2.2.5.2.
Web Feature Service	S. Abschnitt 2.2.5.2.
Linked Open Data (LOD)	Der Begriff <i>Linked Open Data</i> (LOD) beschreibt ein Verfahren, um strukturierte Daten zu veröffentlichen und diese mit anderen LOD zu verknüpfen. Zur Kodierung und Verlinkung der Daten wird RDF und darauf aufbauende Standards wie SPARQL und OWL verwendet, sodass LOD gleichzeitig Teil des Semantic Web ist.
OPC UA	Die <i>OPC Unified Architecture</i> (OPC-UA) stellt die Weiterentwicklung des <i>Object Linking and Embedding (OLE) for Process Control</i> Standards dar. OPC-UA unterstützt die Prozesskommunikation in strukturierter Art und Weise mit einem unterlagerten Informationsmodell [Mahnke, 2009]. OPC-UA basiert auf Web-Technologien und implementiert eine serviceorientierte Client-Server-Architektur. Dadurch ist OPC-UA betriebssystem-, plattformunabhängig und internetfähig.

5.5.3 Zwischenspeicher

Die Komponente Zwischenspeicher lauscht auf *Topics* (Datenkanäle) und schreibt diese Daten und passende Metadaten in einen Datenbank-Cache. Der Cache kann aus laufenden Instanzen von Algorithmen zugegriffen

werden, die in einem Verarbeitungsdienst gehostet werden. Als Cache kann eine der gängigen relationalen Datenbanken, wie zum Beispiel ORACLE, Microsoft SQL Server oder PostgreSQL eingesetzt werden.

5.5.4 Datenarchiv

Je nachdem, ob ein BigData Problem vorliegt (s. Prüfung in der Systemsicht), kann eine gängige relationale Datenbank oder alternativ ein NoSQL Speicher eingesetzt werden. NoSQL Speicher gibt es mittlerweile in großer Anzahl mit den unterschiedlichsten Vor- und Nachteilen. Im Folgenden werden deshalb nur exemplarisch zwei bekannte Vertreter vorgestellt.

5.5.4.1 Apache HDFS und HBase

HDFS ist die Abkürzung für *Hadoop*⁷ *File System* und ist eine Open-Source-Implementierung der Prinzipien hinter dem Google File System (GFS). Apache Hadoop ist ein Entwicklungsrahmen für die BigData Verarbeitung. HDFS ist ein verteiltes Dateisystem, das entwickelt wurde, um auf günstiger Hardware zu laufen, bei der Ausfälle mit hoher Wahrscheinlichkeit auftreten. Um dieses Problem zu umgehen, arbeitet HDFS mit Datenreplikation und schneller Fehlererkennung inkl. automatischer Wiederherstellung. *Apache HBase* ist ein frei verfügbarer, verteilter und spaltenorientierter Speicher nach dem Vorbild von *Google BigTable*. Neben der Java-API bietet HBase auch eine Thrift-Schnittstelle⁸. Thrift ist eine sprachunabhängige Schnittstelle, es gibt jedoch Implementierungen für die gängigsten Programmiersprachen.

5.5.4.2 Apache Cassandra

Cassandra⁹ wurde ursprünglich von Facebook entwickelt als eine Kombination der verteilten Techniken *dynamo* von Amazon und dem *BigTable*

⁷ <http://hadoop.apache.org/>, letzter Abruf am 4.9.2016

⁸ <http://thrift.apache.org/>, letzter Abruf am 4.9.2016

⁹ <http://cassandra.apache.org/>, letzter Abruf am 4.9.2016

Datenmodell von Google. Cassandra nutzt Thrift als Schnittstelle. Da Thrift keine Streaming-Fähigkeiten bietet, muss jedes Objekt in den Speicher passen. Deshalb ist die maximale Größe von Objekten durch die verfügbare Speichergröße begrenzt.

5.5.5 Wissensbasis

Zur Arbeit mit einer Ontologie wird ein sogenannter Ontologiespeicher (engl. *Ontology Store* oder *Triple Store*) benötigt. Ein Leistungsvergleich gängiger Triple Stores wurde vom W3C erstellt [W3C, 2016]. Zwei bekannte Vertreter sind Jena (bzw. Fuseki) und OpenLink Virtuoso. Apache Jena¹⁰ ist ein Open-Source-Framework für die Entwicklung von Semantic Web-Anwendungen. Es basiert auf der Programmiersprache Java und verwendet ein Graphenmodell zur Darstellung von Ontologien. Jena unterstützt RDF, RDFS, OWL und SPARQL. Zur Speicherung des RDF Graphen werden relationale Datenbanken oder diverse Textformate unterstützt. Eine regelbasierte Inferenzmaschine ist für das Reasoning vorhanden, wobei andere Reasoner über eine Schnittstelle angebunden werden können. Das Unterprojekt Fuseki verwendet Jena, um einen SPARQL Endpunkt über HTTP zu implementieren. OpenLink Virtuoso ist ein skalierbarer Cross-Plattform-Server, der Funktionalitäten aus Relationalen-, Graph-, und Dokument-Datenbanken vereint. Der Server ist als Open Source und als kommerzielle Version verfügbar. Als semantische Technologien werden RDF und SPARQL unterstützt.

5.5.6 Verarbeitungsdienste und Simulationen

Auf mögliche Implementierungen wird in Abschnitt 6.3 und 7.3 eingegangen.

¹⁰ <http://jena.apache.org/>, letzter Abruf am 4.9.2016

5.5.7 Verbreitung von Warnmeldungen

Als Übertragungsformat hat sich in den letzten Jahren das OASIS Common Alerting Protocol (CAP) etabliert, das in vielen EWS eingesetzt wird (s. [Meissen, 2014b]). CAP ist ein XML-basiertes Datenformat zum Austausch von Warnmeldungen zwischen Alarmierungstechnologien. CAP ermöglicht es, eine Warnmeldung gleichzeitig an mehrere Warnsysteme zu versenden [Botterell, 2006]. Abbildung 56 zeigt die Datenelemente von CAP. Interessant ist die Berücksichtigung einer Wahrscheinlichkeit (engl. Certainty) und einem Verfallsdatum (engl. Expiration Date) für die Meldung.

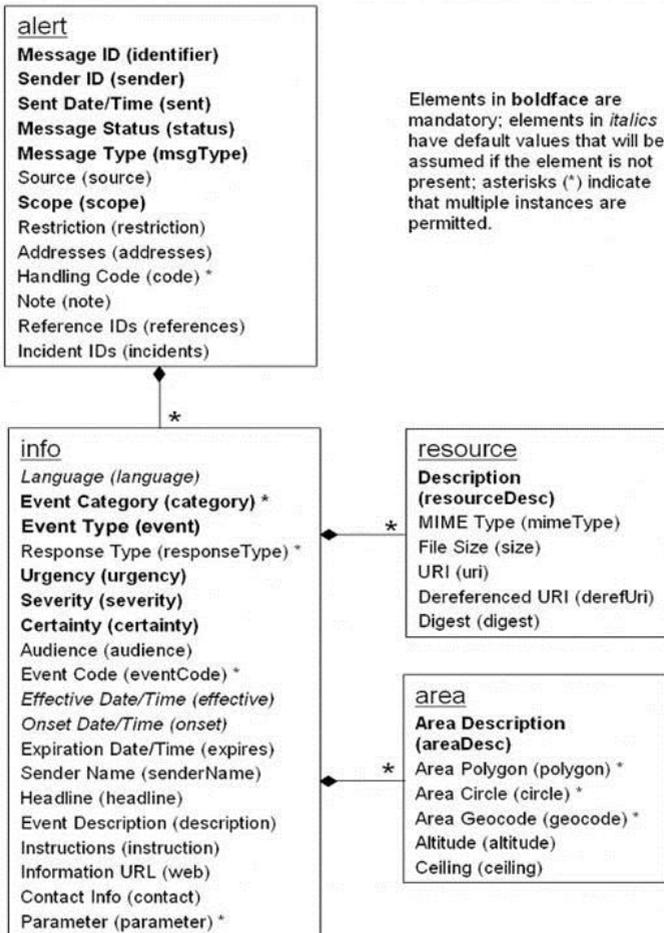


Abbildung 56: Datenelemente von CAP

5.5.8 Workflow-Management-System

Auf Implementierungsmöglichkeiten inkl. Regeln wird in Kapitel 1 eingegangen.

5.5.9 Semantischer Katalog

Auf die Umsetzung des semantischen Katalogs wird in Abschnitt 6.2 eingegangen.

5.5.10 Geographisches Informationssystem

Im kommerziellen Bereich sind hauptsächlich die Produkte der Firma ESRI¹¹, wie zum Beispiel ArcGIS zu nennen. Eine sehr flexible Open-Source Lösung ist die Software *GeoServer*¹². Mit GeoServer können geographische Daten als Web-Services (Geodienste) in einem Netzwerk angeboten werden. Der Server unterstützt diverse Standards insbesondere die Dienste des OGC. Die Darstellung erfolgt beispielsweise mithilfe des WMS oder WCS Dienstes in Form von Grafiken oder es werden mithilfe von WFS die eigentlichen Merkmale zum Beispiel als GML übertragen. Weiterhin gibt es offene Schnittstellen zur Integration eigener Datenquellen.

5.5.11 Benutzerschnittstelle

Um die Anforderungen an die Benutzerschnittstelle erfüllen zu können, sollte ein gängiges Web Content Management System (WCMS) als Basis gewählt werden. Ein WCMS ist eine Software zur gemeinschaftlichen Erstellung, Bearbeitung und Organisation von Inhalten (Content) zumeist in Webseiten, aber auch in anderen Medienformen. Durch den Einsatz eines WCMS wird u.a. die Integration von Dokumenten zur Recherche und die Anbindung an mobile Apps ermöglicht. Das WCMS sollte bereits eine weitreichende Unterstützung für Grafiken, Tabellen und die Integration von Karten enthalten. Weiterhin wäre eine Visualisierung und Navigation über Ontologigraphen wünschenswert. In Kapitel 1 wird die praktische Nutzung eines WCMS in Frühwarnsystemen gezeigt.

¹¹ <http://www.esri.com/>, letzter Abruf am 4.9.2016

¹² <http://geoserver.org/>, letzter Abruf am 4.9.2016

5.5.12 Durchzuführende Schritte

Folgende in Tabelle 31 aufgeführten Schritte müssen zur vollständigen Ausarbeitung dieser Sicht durchgeführt werden.

Tabelle 31: Arbeitsschritte der Technologiesicht

#	S. Abschnitt	Arbeitsschritt
1	5.5	Analyse der aktuell verfügbaren Hard- und Softwarelösungen.
2	5.5.1 - 5.5.11	Auswahl der passenden Softwarelösungen für jede Komponente.

5.6 Konstruktionsicht

Die Konstruktionsicht bildet die Systemsoftware auf eine konkrete Laufzeitumgebung ab.

5.6.1 Interaktionsmatrix und Integrationstests

Die folgende Tabelle 32 zeigt die Komponenten und Dienste von RA-FWS^{NG} und deren Wechselwirkungen. Die Komponenten in der ersten Spalte sind die Akteure oder Absender und verwenden somit die Schnittstelle der Komponenten in der ersten Zeile. Eine Komponente, die die Schnittstelle einer anderen Komponente verwendet, wird in grüner Farbe in der entsprechenden Zelle markiert.

Tabelle 32: Integrationsmatrix RA-FWS^{NG} mit Testfällen

Empfänger ► Sender ▼	MOM	Daten- quellen	Semant. Katalog	Zwischen- speicher	Daten- archiv	Wissens- basis	Verarbeit- ungsdienste	Workflow- M.-System	Verbreitung Meldungen	Benutzer- schnittstelle
MOM			P3-1	P1-1	P3-2			P2-1		P2-2
Datenquellen	P1-2		P2-3	P2-4	P3-3		P1-3			P1-8
Semantischer Katalog										
Zwischen- speicher	P3-4									
Datenarchiv										
Wissensbasis										
Verarbeit- ungsdienste	P2-7			P1-5		P1-10		P1-4		P1-9
Workflow- M.- Dienst	P1-6		P2-5	P3-5		P1-11	P1-7		P2-8	P2-6
Verbreitung Meldungen										
Benutzer- schnittstelle			P3-6		P3-7			P3-8		

Da die meisten der Interaktion zwischen den Komponenten über die MOM entkoppelt sind, wird eine solche Interaktion in zwei Zellen gekennzeichnet, die Zelle der MOM und die Zelle der Empfangskomponente. Für diese Wechselwirkungen ist ein dreistufiger Test notwendig:

1. Kann die Senderkomponente eine Nachricht an die MOM schicken?
2. Erhält die Empfängerkomponente die Nachricht von der MOM?
3. Kann die Nachricht von der Empfängerschnittstelle in Bezug auf Syntax und Semantik verarbeitet werden?

Es ist empfehlenswert, die Integration in mehreren Schritten vorzunehmen. Die Integrationstests sind deshalb in drei Phasen aufgeteilt. Die Kennung in einer Zelle entspricht der spezifischen Phase (P1-P3), in der diese Schnittstelle integriert wird, ergänzt um eine eindeutige Nummer für den Testfall. Im endgültigen System müssen alle Testfälle ausgeführt werden. Tabelle 33, Tabelle 34 und Tabelle 35 zeigen die Testfälle der drei Phasen.

Tabelle 33: Integrationstests Phase 1

P1-1	Der Zwischenspeicher empfängt Daten von der MOM und speichert sie im Cache.
P1-2	Datenquellen senden Daten über die MOM. Es muss einen Test für jede spezifische Quelle vorhanden sein, die im Zusammenhang mit dem Anwendungsfall steht.
P1-3	Verarbeitungsdienste verarbeiten (Echtzeit-)Daten aus den Datenquellen. Diese müssen wiederum für jeden Algorithmus getestet werden.
P1-4	Das Ergebnis eines Verarbeitungsdienstes wird durch einen spezifischen Workflow im Workflow-Management-System verwendet (dies ist spezifisch für Anwendungsfälle).
P1-5	Verarbeitungsdienste greifen auf den Cache zu um Daten zu lesen.
P1-6	Ein Workflow sendet ein Ergebnis auf die MOM.
P1-7	Ein Workflow ruft einen Verarbeitungsdienst auf.
P1-8	Die Benutzeroberfläche empfängt und zeigt Sensordaten an.

P1-9	Die Benutzeroberfläche empfängt und zeigt die Ergebnisse von Verarbeitungsdiensten an.
P1-10	Ein Verarbeitungsdienst schreibt sein Ergebnis in die Wissensbasis.
P1-11	Ein Verarbeitungsdienst nutzt Daten aus der Wissensbasis.

Tabelle 34: Integrationstests Phase 2

P2-1	Der Workflow-Management-Service empfängt eine Nachricht von der MOM, um einen neuen spezifischen Workflow zu instanziiieren.
P2-2	Die Benutzeroberfläche zeigt Meldungen (Ereignisse) aus der MOM an.
P2-3	Der semantische Katalog behandelt Registrierungsanforderungen von neuen Datenquellen.
P2-4	Der Zwischenspeicher empfängt (spezifische) Daten von der Sensorik.
P2-5	Ein spezifischer Workflow fragt beim semantischen Katalog zusätzliche Informationen (zum Beispiel verfügbare Sensorik oder Simulationen) an.
P2-6	Ein spezifischer Workflow sendet Nachrichten, die auf der Benutzeroberfläche angezeigt werden.
P2-7	Ein Verarbeitungsdienst sendet sein Ergebnis auf die MOM.
P2-8	Ein spezifischer Workflow sendet eine Warnmeldung.

Tabelle 35: Integrationstests Phase 3

P3-1	Der semantische Katalog empfängt Anforderungen zur Registrierung neuer Datenquellen über die MOM.
P3-2	Der Speicher empfängt Daten von der MOM (Syntax).
P3-3	Der Speicher empfängt Daten von den Datenquellen (Semantik).
P3-4	Der Zwischenspeicher sendet Daten auf die MOM.
P3-5	Ein spezifischer Workflow fordert die Cache-Speicherung von Daten beim Zwischenspeicher an.
P3-6	Die Benutzeroberfläche fragt den semantischen Katalog nach verfügbaren Datenquellen (zum Beispiel zur Anzeige auf einer Karte).
P3-7	Die Benutzeroberfläche zeigt Daten aus dem Speicher.
P3-8	Die Benutzeroberfläche löst einen Workflow aus.

5.6.2 Verteilung der Komponenten

Die Verteilung der Komponenten (engl. Deployment) auf Rechnersysteme und Netzwerke ist spezifisch für jedes konkrete System und kann deshalb an dieser Stelle nicht näher ausgeführt werden.

5.6.3 Sicherheit in verteilten Systemen

Sicherheitsaspekte in verteilten Systemen stellen ein eigenes Forschungsgebiet dar und können an dieser Stelle nicht in voller Tiefe behandelt werden. Deshalb werden im Folgenden die wichtigsten Probleme stichpunktartig angesprochen:

1. Absicherung des Zugriffs auf das System
2. Unterstützung verschiedener Rechte von Benutzern bzw. Rollen
3. Absicherung des Netzwerks
4. Punkt 1 bis 3 unter dem Aspekt, mehrere Standorte zu unterstützen

Zu den Punkten 1 und 2 bringen WCMS, bereits Lösungen mit. Müssen mehrere Standorte verbunden werden (Punkt 4), ist eine gemeinsame Benutzer- und Rechteverwaltung zu empfehlen und ggf. auch eine *Single Sign-on* Lösung zu implementieren. Auf dem Markt sind bereits vielfältige Softwarelösungen erhältlich, die dazu eingesetzt werden können.

Zur Absicherung des Netzwerks ist die Ideallösung ein komplett vom Internet getrenntes Netz zu betreiben. Sollte dies nicht möglich sein, kann auf ein virtuelles, privates (in sich geschlossenes) Kommunikationsnetz (VPN, engl. Virtual Private Network) gesetzt werden. Drahtlose Kommunikation sollte nach Möglichkeit vermieden werden, um das einfache Mitschneiden des Netzwerkverkehrs zu verhindern. Weiterhin sollte der gesamte Netzwerkverkehr verschlüsselt werden. Zur standortübergreifenden Kopplung werden die Broker-Cluster, die die MOM bilden, verbunden. Diese ermöglichen eine Vertrauensstellung durch gegenseitige Authentifizierung (s. [Tao, 2012]).

5.6.4 Durchzuführende Schritte

Die in Tabelle 36 aufgeführten Schritte müssen zur vollständigen Ausarbeitung dieser Sicht für ein konkretes Frühwarnsystem durchgeführt werden.

Tabelle 36: Arbeitsschritte der Konstruktionssicht

#	S. Abschnitt	Arbeitsschritt
1	5.6.1	Spezifikation der spezifischen Integrationstests für verwendeten Datenquellen, Verarbeitungsdienste, Simulationen und Workflows.
2	5.6.1	Ausführung aller Integrationstests.
3	5.6.2	Spezifikation der Verteilung der Komponenten auf Rechnersysteme und Netzwerke.
4	5.6.3	Ausarbeitung des Sicherheitskonzepts.

5.7 Gesammelte Arbeitsschritte

Tabelle 37 zeigt alle auszuführenden Arbeitsschritte bei der Erstellung einer konkreten Architektur für ein Frühwarnsystem unter Verwendung von RA-FWS^{NG}.

Tabelle 37: Durchzuführende Arbeitsschritte für RA-FWS^{NG}

#	S. Abschnitt	Arbeitsschritt
Unternehmenssicht		
1	5.1.1, 5.1.2, 5.1.3	Analyse der Hauptakteure, Rollen und ihrer Interaktionen.
2	5.1.4.1	Identifikation der Frühwarnindikatoren.
3	5.1.4.1	Identifikation der mathematischen Modelle und Analyseverfahren.
4	5.1.4.2	Festlegung der Anwendungsfälle mit Hilfe von SOPs.

System von Systemen Sicht		
5	5.2.1	Prüfung, ob ein SoS vorliegt.
6 (opt.)	5.2.2	Ggf. Implementierung einer skalierbaren und resilienten Kommunikationsschicht.
Systemsicht		
7	5.3.9	Umsetzung der in der Unternehmenssicht definierten SOPs.
8	5.3.2	Integration aller notwendigen Datenquellen.
9	5.3.6	Anbindung und Implementierung aller Verarbeitungsdienste.
10	5.3.6.4	Prüfung, ob ein BigData Problem vorliegt.
11 (opt.)	5.3.6.4	Integration BigData Verarbeitung.
12	5.3.7	Anbindung und Implementierung aller Simulationen.
13	5.3.11	Identifikation und Integration des benötigten Kartenmaterials.
14	5.3.12	Spezifikation der Benutzeroberfläche.
Informationssicht		
15	5.4.1	Erstellung der SensorML Beschreibungen für die spezifischen Sensortypen.
16	5.4.4 / 6.1	Integration von Domänen-Ontologien.
Technologiesicht		
17	5.5	Analyse der aktuell verfügbaren Hard- und Softwarelösungen.
18	5.5.1 - 5.5.11	Auswahl der passenden Softwarelösungen für jede Komponente.
Konstruktionssicht		
19	5.6.1	Spezifikation der spezifischen Integrationstests für verwendeten Datenquellen, Verarbeitungsdienste, Simulationen und Workflows.
20	5.6.1	Ausführung aller Integrationstests.
21	5.6.2	Spezifikation der Verteilung der Komponenten auf Rechnersysteme und Netzwerke.

6 Einsatz semantischer Technologien in Frühwarnsystemen

Semantik spielt in den verschiedensten Komponenten eines Frühwarnsystems eine Rolle. Semantische Technologien eignen sich zur Lösung verschiedener Architekturprobleme, wie zum Beispiel für die semantische Modellierung der Domäne, die Verwaltung der Sensorik, die Zusammenführung heterogener Daten, die Speicherung des Wissens, die Navigation durch das Informationsnetz und die Präsentation von Informationen. Abbildung 57 zeigt, an welchen Stellen der generischen Architektur RA-FWS^{NG} semantische Technologien zum Einsatz kommen.

- *Semantik 1 – Datenintegration:* Die heterogenen Daten der Sensoren müssen zur Auswertung in eine Datenbasis integriert werden. Dazu können semantische Annotationen eingesetzt werden.
- *Semantik 2 – Verwaltung von Sensorik und Diensten:* In einem verteilten System müssen Sensoren, Verarbeitungs- und Simulationsdienste dynamisch hinzugefügt, entfernt und aufgefunden werden können.
- *Semantik 3 – Verarbeitungsdienste:* Zur Datenübergabe zwischen Verarbeitungsdiensten eignen sich Ontologien. Weiterhin kann ein Dienst semantische Abfragen zur Datenselektion einsetzen.
- *Semantik 4 – Wissensbasis:* Die Speicherung des Wissens erfolgt mittels einer Ontologie, die passend zur Domäne modelliert wurde.
- *Semantik 5 – Verbreitung von Warnmeldungen:* Personalisierte und mehrsprachige Warnmeldungen können mittels Ontologien und Regeln generiert werden.
- *Semantik 6 – Benutzerschnittstelle:* Zum besseren Situationsverständnis können Benutzer durch das Informationsnetz der Wissensbasis navigieren und semantische Anfragen stellen.

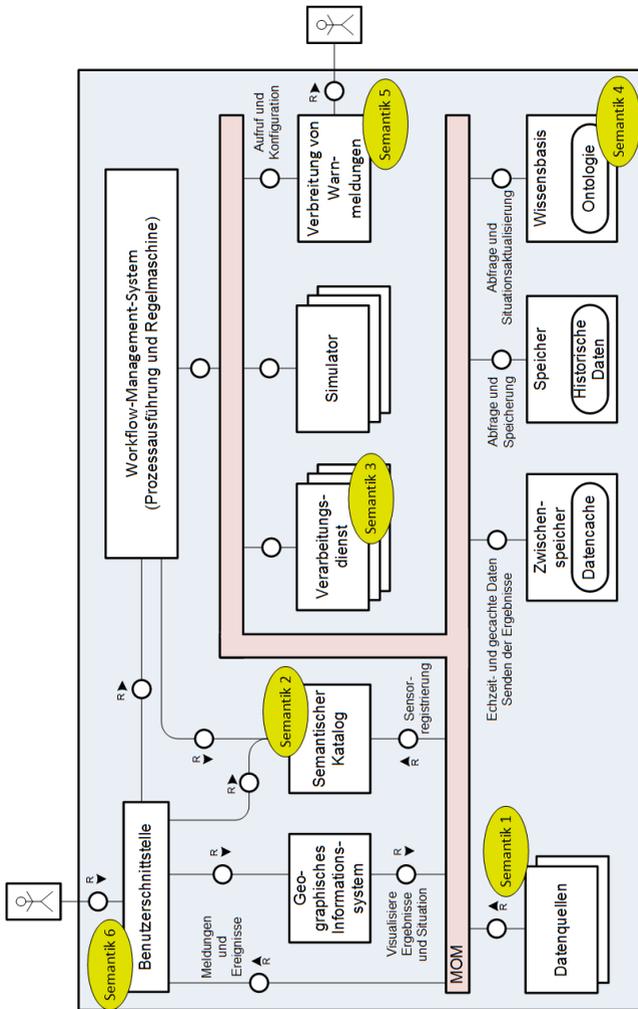


Abbildung 57: Einsatz semantischer Technologien in RA-FWS^{NG}

6.1 Datenintegration

Die heterogenen Daten der Sensoren müssen zur Auswertung in eine Datenbasis integriert werden. Die unterschiedlichen Datenmodelle müssen dazu auf ein homogenes Datenmodell abgebildet (Transformation) oder alternativ mit entsprechenden semantischen Annotationen versehen werden (Analyse). Im Bereich der Abbildung von (Daten-) Modellen wurde bereits viel geforscht. Insbesondere die Abbildung der Schemata von relationalen Datenbanken, die bereits seit Langem verfügbar sind, wurde untersucht. Einen guten Überblick über den Stand der Forschung in diesem Bereich geben Bellahsene, Bonifati und Rahm [Bellahsene, 2011]. Neuere Forschung konzentriert sich auf XML und Ontologie-Modelle [Gross, 2012], von denen die letzteren für die Abbildung den Vorteil besitzen, dass die Semantik des Modells zur Verfügung steht. Darüber hinaus werden Abbildungen zwischen diesen verschiedenen Arten von Modellen erforscht. Bisher gibt es jedoch keinen vollautomatischen Abbildungsalgorithmus, der das Problem vollständig löst [Bernstein, 2011].

Bügel et al. zeigen als Alternative ein semi-automatisches Vorgehen zur Zusammenführung von Umweltdaten [Bügel, 2011]. In diesem Konzept werden in mehreren iterativen Schritten die automatisch berechneten Abbildungen von einem Fachexperten überarbeitet, die dann wieder als verbesserte Eingabe für den Berechnungsalgorithmus bilden.

Im EU Projekt ENVIROFI wurde festgestellt, dass eine Integration in ein gemeinsames Datenmodell zwar wünschenswert, aber unrealistisch ist. Stattdessen wird vorgeschlagen, mit kleineren Teilmodellen zu arbeiten, die auf einen bestimmten Anwendungsaspekt zugeschnitten sind [ENVIROFI, 2012]. Im Projekt TRIDEC wurde zur Adressierung des Problems ein ähnliches Vorgehen gewählt. Die jeweiligen zu nutzenden Daten wurden mit semantischen Annotationen versehen, aber weiterhin im Originalformat beibehalten [Moßgraber, 2012a].

6.1.1 Datenveröffentlichung und -transformation

Für die Integration heterogener Datenquellen sind folgende Schritte auszuführen, die auch in Abbildung 58 dargestellt sind.

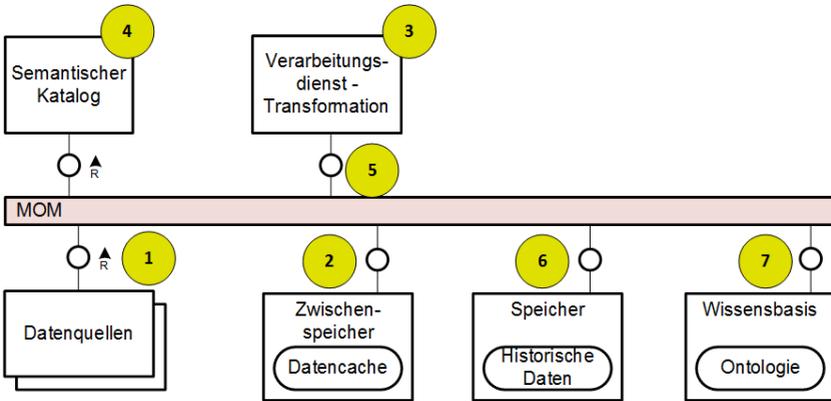


Abbildung 58: Transformations- und Analyseprozess zur Datenintegration

1. Codieren domänenspezifischer Datenmodelle unter Verwendung von einem oder mehreren offenen Datenformaten (beispielsweise RDF). Die Schemata und Vokabulare dieser Datenmodelle verwenden etablierte, domänenspezifische Standards. Dies ermöglicht es, Datenquellenanbietern einer Domäne, Daten ohne größeren Aufwand (auf der MOM) zu veröffentlichen und ihre eigenen vertrauten Datenmodelle zu verwenden.
2. Ein Analyse- oder Transformationsdienst abonniert sich auf die Datenquelle.
3. Je nach Anwendungsfall transformiert der Dienst das Datenmodell oder fügt alternativ semantische Annotationen hinzu. Zur Erstellung der Abbildung für die Transformation kann das von Bügel

beschriebene semi-automatische Verfahren verwendet werden [Bügel, 2011].

4. Ggf. wird der semantische Katalog angefragt, um Informationen über die Sensorik zu erhalten, wie beispielsweise die Einheiten der Messwerte. Für den Transformationsdienst sollten Basis-Transformationen zur Verfügung stehen, die häufig benötigt werden (Transformation von Koordinatensystemen oder Einheiten).

Die folgenden Punkte sind als Alternativen zu sehen.

5. Der Verarbeitungsdienst veröffentlicht die transformierten Daten oder die Annotationen auf der MOM.
6. Der Verarbeitungsdienst schreibt die transformierten Daten in den Datenspeicher.
7. Der Verarbeitungsdienst schreibt die Annotationen in die Wissensbasis.

6.1.2 Semantische Integration von Informationssystemen

Wie in Kapitel 1 beschrieben, müssen neben Sensoren auch Daten existierender Informationssysteme in ein Frühwarnsystem integriert werden können. D.h. die Schnittstelle eines Informationssystems muss für ein neues Schnittstellenprotokoll (oder mehrere), wie MOM oder OGC SOS, ertüchtigt werden. Abbildung 59 verdeutlicht die Problematik.

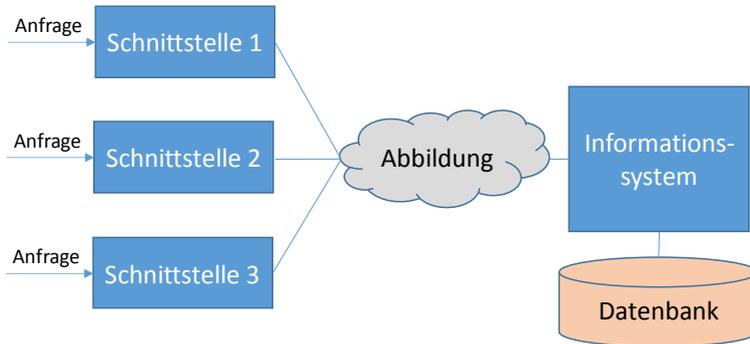


Abbildung 59: Abbildung mehrerer Schnittstellen auf ein existierendes Informationssystem

Eine Lösung mit Hilfe einer Ontologie kann mit folgenden Schritten umgesetzt werden:

- Für jede Schnittstelle, die unterstützt werden soll, wird eine Ontologie benötigt. Die Ontologie muss die spezifischen Konzepte und Eigenschaften enthalten, um das Modell zu beschreiben. Vorzugsweise sollte eine bestehende Ontologie wiederverwendet werden. Im Rahmen von RA-FWS^{NG} sollte die SSNO eingesetzt werden.
- Alle erforderlichen Konzepte und die zugehörigen Eigenschaften, die in der verwendeten Ontologie enthalten sind, werden auf das Datenmodell des Informationssystems abgebildet. Dies wird durch das Hinzufügen von Annotationen umgesetzt, die aus den Ressourcen URIs der Ontologie bestehen. Die verfügbaren Datensätze werden dann beispielsweise als SOS-Beobachtungen mit #Observation annotiert. Die Attribute der Datensätze werden mit Eigenschaften aus den Ontologien annotiert, zum Beispiel #hasValue, #hasTime.
- Die zu erzeugende Struktur ergibt sich aus der Struktur der Ontologie. Entsprechend verknüpfte Datensätze werden ebenfalls durch die Relationen der Ontologie erkannt.

- Die Umsetzung der Datensätze erfolgt schließlich über einen zwischengeschalteten Dienst, der Kenntnis von den Annotationen und der Ontologie hat.

Eine detaillierte Beschreibung der Vorgehensweise, die in dieser Arbeit zu weit führen würde, wird in *Automating the web publishing process of environmental data by using semantic annotations* anhand eines Realbeispiels gezeigt [Moßgraber, 2014].

6.2 Verwaltung von Sensorik und Diensten

In einem verteilten System müssen Sensoren, Verarbeitungs- und Simulationsdienste dynamisch hinzugefügt, entfernt und aufgefunden werden können. Die *Universal Description Discovery and Integration* (UDDI) Spezifikation wurde als Katalogdienst für Service-orientierte Architekturen vorgeschlagen [Bellwood, 2002]. Die Fähigkeiten zur Nutzung zum Auffinden von Diensten sind eher begrenzt, da eine maschinenverständliche Semantik in den technischen Spezifikationen und Klassifikationen nicht vorgesehen ist. Der Standard scheiterte an der mangelnden Akzeptanz bei großen Industrieunternehmen und muss als obsolet eingestuft werden. Ein neuer Katalog wurde im Rahmen des EU Projekts FUSION entwickelt, um die Beschränkungen von UDDI zu beheben [Bouras, 2010]. Der Katalog arbeitet mit semantischen Beschreibungen der von ihm verwalteten Dienst-schnittstellen mittels OWL-DL. Für die Abfrage und das Auffinden von Diensten wird logisches Schließen (engl. Description Logic Reasoning) eingesetzt [Kourtesis, 2009].

Der Abschlussbericht der W3C Semantic Sensor Network Incubator Group beschreibt einen ähnlichen semantischen Ansatz für die Integration von Diensten in der Sensornetzdomäne, die an ähnlichen Mängeln leidet [Lefort, 2011]. Obwohl in diesem Bereich schon diverse Einzelarbeiten durchgeführt wurden (beispielsweise zur ontologischen Repräsentation des OGC O&M Modells) stellt der Bericht die erste umfassende Studie zum Einsatz

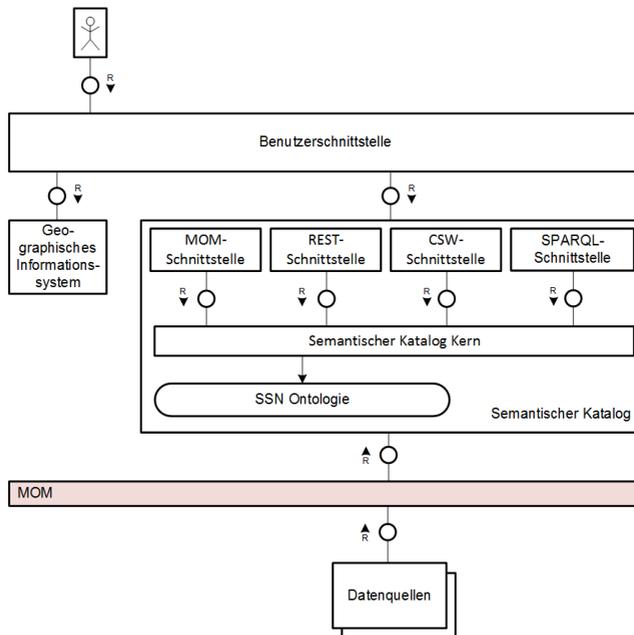
semantischer Technologien für Sensornetzanwendungen dar. Der Bericht gibt Empfehlungen zu zwei Hauptpunkten ab: 1) die Entwicklung einer Ontologie zur Beschreibung von Sensoren und Sensornetzen für den Einsatz in Sensor Web-Anwendungen und 2) die Empfehlung von Methoden, wie die Ontologie innerhalb von semantischen Anwendungen mit den OGC SWE Standards verwendet werden kann. Ein weiterer Ansatz für einen semantischen Katalog wurde von dem EU Projekt SemSorGrid4Env unter dem Namen Strabon (stSPARQL-enabled Registry for Data Sources) entwickelt [Gray, 2010]. Der Katalog unterstützt das Auffinden von Sensoren, Sensornetzwerken, Diensten und Datenquellen bei entsprechender Beschreibung. Diese Implementierung basiert auf einer proprietären Erweiterung von RDF (genannt stRDF), die für jede Ressource räumliche und zeitliche Metadaten erlaubt und einer Abfragesprache, die SPARQL ebenfalls mit räumlich und zeitlichen Fähigkeiten (genannt stSPARQL) erweitert.

Xiao et al. beschreiben den Prototypen *Sensor Semantic Registry Service (SSRS)*, bei dem sie auf die OGC CSW Katalogschnittstelle aufbauen [Xiao, 2016]. Für die semantische Erweiterung des Katalogs wurde eine eigene Ontologie entwickelt mit dem Namen *Sensor Metadata Ontology (SMO)*. Als Vorteil gegenüber SSNO wird die Unterstützung von Sensoren in Satelliten und die besseren Möglichkeiten bei der Registrierung und Verwaltung von Sensordatenquellen angeführt. Jedoch ist dafür die Ausdrucksstärke für die Beschreibung der Sensorik niedriger.

6.2.1 Schnittstellen zum semantischen Katalog

Abbildung 60 zeigt den Aufbau des semantischen Katalogs, zur Verwaltung der Sensorik und Dienste und seine Integration in RA-FWS^{NG}. Zur flexiblen Integration in Frühwarnsysteme und den Zugriff von den unterschiedlichsten Komponenten werden mehrere Schnittstellen unterstützt.

1. REST-Schnittstelle: Über die REST-Schnittstelle können Sensoren und Dienste registriert und abgerufen werden. Außerdem können zusätzliche Informationen zu einem bestimmten Sensor und den zugehörigen Datenquellen (*MOM Topics*) angefragt werden.
2. MOM-Schnittstelle: Unterstützt die gleichen Funktionen wie die REST-Schnittstelle.
3. CSW-Schnittstelle: Die Schnittstelle des OGC Catalogue Service unterstützt zwar keine semantischen Anfragen, jedoch wird aus Gründen der Rückwärtskompatibilität auch diese Schnittstelle unterstützt.
4. SPARQL-Schnittstelle: Nimmt lesende Anfragen über einen http-Endpoint entgegen. Neue Sensoren und Dienste können nicht registriert werden.

Abbildung 60: Aufbau des semantischen Katalogs und Integration in RA-FWS^{NG}

Über die MOM- und die REST-Schnittstelle können die in Tabelle 38 aufgeführten Befehle verwendet werden.

Tabelle 38: Befehlssatz der MOM- und REST-Schnittstelle

registerSensor	Registriert einen Sensor mittels einer SensorML Beschreibung.
listSensors	Liefert alle registrierten Sensoren.
describeSensor	Liefert die SensorML Beschreibung eines Sensors.
deleteSensor	Löscht einen Sensor aus dem Katalog.
updateSensor	Aktualisiert die Beschreibung eines Sensors.
registerService	Registriert einen Dienst mittels einer SensorML Beschreibung des Prozesses.
listServices	Liefert alle registrierten Dienste
describeService	Liefert die SensorML Beschreibung eines Dienstes.
deleteService	Löscht einen Dienst aus dem Katalog.
sparqlQuery	Führt eine SPARQL-Anfrage auf dem Datenbestand aus.

6.2.2 Implementierung des semantischen Katalogs

Die im Kapitel 1 gezeigten Anwendungsbeispiele von RA-FWS^{NG} verwenden einen semantischen Katalog, der auf dem WCMS WebGenesis[®] aufsetzt [Chaves, 2013]. Das System wurde am Fraunhofer IOSB¹ entwickelt und unterstützt im Gegensatz zu anderen WCMS Ontologien. Dadurch konnte durch Import der SSNO sehr schnell ein semantischer Katalog erstellt werden. Implementiert werden mussten nur noch die spezifischen MOM-, REST- und CSW-Schnittstellen.

Abbildung 61 zeigt die SSNO, die in ein WebGenesis[®] System importiert wurde. Die von der SSNO referenzierten Konzepte und Eigenschaften aus der DUL Ontologie wurden ebenfalls importiert. Im System kann durch die Hierarchie der Ontologie und ihre Eigenschaften navigiert werden.

¹ <http://www.iosb.fraunhofer.de/?18052>, letzter Abruf am 4.9.2016

Abbildung 62 zeigt die Darstellung des SSNO Konzepts *Sensor* mit seinen Eigenschaften (in der Abbildung sind nicht alle Eigenschaften sichtbar).

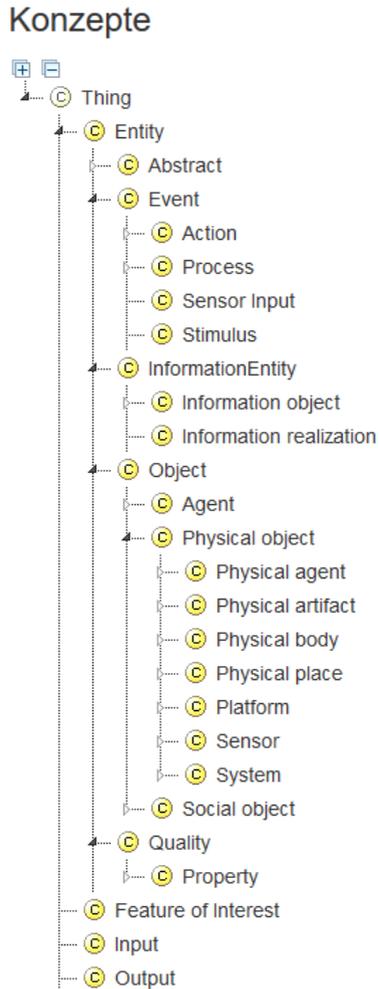


Abbildung 61: In WebGenesis® importierte Konzepte aus SSNO und DUL

Sensor

Basisklasse(n):PhysicalObject

Unterklasse(n):Sensing Device

A sensor can do (implements) sensing: that is, a sensor is any entity that can follow a sensing method and thus observe a Property of a FeatureOfInterest. Sensors may be physical devices, computational methods, a laboratory setup with a method, or any other thing that can follow a Sensing Method to observe a Property.

Objekt-Eigenschaften

deployed on platform

Thing *deployed on platform* Platform

Relation between a deployment and the platform on which the system was deployed.

deployed system

Thing *deployed system* System

Relation between a deployment and the deployed system.

deployment process part

Thing *deployment process part* Deployment-related Process

Has part relation between a deployment process and its constituent processes.

describes

Thing *describes* Thing

detects

Thing *detects* Stimulus

A relation from a sensor to the Stimulus that the sensor can detect.

The Stimulus itself will be serving as a proxy for (see isProxyOf) some observable property.

end time

Thing *end time* Thing

feature of interest

Thing *feature of interest* Feature of interest

A relation between an observation and the entity whose quality was observed. For example, in an observation of the person, the feature of interest is the person and the quality is weight.

for property

Thing *for property* Property

A relation between some aspect of a sensing entity and a property. For example, from a sensor to the properties it can sense, from a deployment to the properties it was installed to observe. Also from a measurement capability to the property described for. (Used in conjunction with ofFeature).

has input

Thing *has input* Input

has measurement capability

Thing *has measurement capability* Measurement Capability

Relation from a Sensor to a MeasurementCapability describing the measurement properties of the sensor.

Abbildung 62: Hierarchie, Beschreibungstext und Eigenschaften des SSNO Konzepts Sensor

Ontologien können in WebGenesis® importiert und aus ihren Konzepten und Eigenschaften können automatisch Eingabeformulare erstellt werden. Weiterhin ist ein Editor für die Erstellung semantischer Suchen enthalten.

Eingabeformular für Sensor

The image shows a web form titled "Eingabeformular für Sensor". It contains five vertically stacked dropdown menus. Each menu has a label above it, a large empty text box with a scroll bar on the right, and a small button labeled "Auswahl" below it. The labels are: "Deployed on platform", "Deployed system", "Deployment process part", "Describes", and "Detects".

Abbildung 63: Automatisch generiertes Eingabeformular für das SSNO Konzept Sensor

Abbildung 63 zeigt einen Ausschnitt des Formulars, das für das Konzept *Sensor* erzeugt wurde.

Sensorsuche

Ontologie

ssn (415) 116 89 2 6

Laden

Semantische Anfrage

Sensor	type	Sensor	-
Sensor	has parameter	Parameter	-
			+

Optionen

Ausgabe Maximale Anzahl von Ergebnissen Anfrage automatisch ausführen

Sensor

Sortierung

Sensor

Speichern Abbruch

Abbildung 64: Editor für semantische Suchen

Zusätzlich ist auch ein Editor enthalten, mit dem Fachanwender semantische Suchen auf der Ontologie erstellen können. Abbildung 64 zeigt ein passendes Beispiel. Die Werte der Variable *Sensor* müssen vom Typ des SSNO Konzepts *Sensor* sein und einen bestimmten Parameter besitzen. Enthält die konfigurierte Anfrage eine oder mehrere ungebundene Variablen (im Beispiel die Variable *Parameter*), werden automatisch Eingabefelder für diese im Suchformular erstellt. Dies ist in Abbildung 65 dargestellt. Sind keine freien Variablen vorhanden, wird das Suchergebnis direkt angezeigt.

Sensorsuche

Parameter*

Suchen Bearbeiten

Abbildung 65: Aus der semantischen Suche generiertes Suchformular

Im System wird aus der konfigurierten Suche eine SPARQL-Anfrage erzeugt. Folgender SPARQL-Code entspricht dem obigen Beispiel:

```

BASE      <http://purl.oclc.org/NET/ssnx/ssn#>
PREFIX rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#
PREFIX dul: http://www.loa-cnr.it/ontologies/DUL.owl#

SELECT DISTINCT ?Sensor
WHERE
{
    ?Sensor    rdf:type                <#Sensor> ;
              dul:hasParameter>    ?Parameter .
}
ORDER BY ASC(?Sensor)

```

Durch diese Funktionalitäten kann sehr schnell eine Datenbasis für die vorhandene Sensorik eines Frühwarnsystems aufgesetzt und die gewünschten Suchanfragen getestet werden.

6.3 Verarbeitungsdienste

Ontologien können von Verarbeitungsdiensten zu unterschiedlichen Zwecken eingesetzt werden:

- Dienste, die auf der Datenebene arbeiten, wie beispielsweise Analyse- und Fusionsalgorithmen liefern ihre Ergebnisse als ontologische Instanzen [Vrochidis, 2012], [Moumtzidou, 2012].
- Dienste, die auf der Wissensebene angesiedelt sind (beispielsweise zur Entscheidungsunterstützung), benutzen die ontologische Wissensbasis als Datengrundlage und liefern ggf. wieder ontologische Instanzen als Ergebnis. Weiterhin kann ein Dienst semantische Abfragen zur Datenselektion einsetzen [Wanner, 2015].
- Dienste können Reasoning verwenden zum Folgern von Schlüssen, beispielsweise zur Erkennung einer Situation oder Gefahr [Pengel, 2013], [Sharma, 2016].

Der interessanteste Anwendungsfall ist die Verwendung von Reasoning zum Folgern von Schlüssen, beispielsweise zur Erkennung einer Situation oder Gefahr. Folgende Regel zeigt dazu ein Beispiel.

```
[ruleAbundantPollen:
(?request rdf:type problem:AnyHealthIssue)
(?request problem:hasUser ?user)
(?user problem:isSensitiveTo ?pollen)
(?pollen rdf:type data:PollenDataType)
(?request :hasGeoArea ?geoArea)
(?request :hasData ?data)
(?data data:hasEnvironmentalDataType ?pollen)
(?data data:hasAggregationType data:max)
(?data data:hasRating ?rating)
(?rating data:hasRatingValue data:abundantPollen)
makeTemp(?rec)
->
(?rec rdf:type conclusions:Recommendation)
(?rec conclusions:hasRecommendationType
conclusions:recommendation_abundantPollen)
(?rec conclusions:hasWeight 1.0)
(?request:hasConclusion ?rec)
(?data :ProduceConclusion ?rec)]
```

Die Regel sagt aus, dass wenn

- eine Person empfindlich gegen Pollen ist,
- in dem Zielgebiet Pollenflug herrscht und
- die Rate einen Grenzwert überschreitet,

eine Warnmeldung erstellt werden muss.

6.4 Wissensbasis

Die Speicherung des Wissens erfolgt mittels einer Ontologie, die passend zur Domäne modelliert wurde.

6.4.1 Analyse verfügbarer Ontologien

Bereits verfügbare Ontologien wurden für RA-FWS^{NG} recherchiert und sind in Tabelle 39 aufgeführt. Diese wurden kategorisiert nach generischen Basisontologien, Ontologien, die einen Raumbezug modellieren, zeitbezogene Ontologien, Ereignis-Ontologien (im Sinne von *etwas passiert*), Ontologien für Sensorik, Ontologien für Sensordaten und spezifische Ontologien für die Frühwarnung und das Katastrophenmanagement.

Tabelle 39: Relevante Ontologien

#	Name und URL	Kommentar
	Basisontologien	
1	DOLCE + DnS Ultralite Ontology (DUL) http://www.ontologydesignpatterns.org/ont/dul/DUL.owl	Organisationen, Beziehungen, Planung, Ereignisse, ...
2	Friend of a Friend (FOAF) http://xmlns.com/foaf/0.1/	Personen, Organi- sationen und ihre Beziehungen

Raum		
3	SWEET (Semantic Web for Earth and Environmental Terminology) https://sweet.jpl.nasa.gov/download	Begriffe aus dem Geo- und Umweltbereich
4	GeoNames http://www.geonames.org/ontology/documentation.html	Orte, Länder, Bevölkerung, Postleitzahl, ...
5	Basic Geo Vocabulary http://www.w3.org/2003/01/geo/	Orte
Zeit		
6	OWL-Time http://www.w3.org/2006/time https://www.w3.org/TR/owl-time/ (Update 2016)	Zeit
7	Time-indexed Situation http://www.ontologydesignpatterns.org/cp/owl/timeindexedsituation.owl	Reihenfolge von Ereignissen
8	The Timeline Ontology http://motools.sourceforge.net/timeline/timeline.html	Reihenfolge von Ereignissen, erweitert OWL-Time
Sensorik und Sensordaten		
9	Sensor and Sensor Network ontology (SSN ontology) https://www.w3.org/2005/Incubator/ssn/ssnx/ssn	Sensorik und Beobachtungen. Kein Raum und Zeit. Basiert auf DUL.
10	Sensor Metadata Ontology (SMO) [Xiao, 2016]	Sensorik, Registrierung und Verwaltung
Ontologien für Ereignisse, Frühwarnung und Katastrophenmanagement		
11	Event Model F http://ontologydesignpatterns.org/wiki/Ontology:Event_Model_F	Raum und Zeit, Objekte, Ursache und Wirkung; erweitert DUL
12	Flood ontology network http://www.semsorgrid4env.eu/index.php/ontologies/13-flood-ontology-network	Integriert DUL, SWEET, FOAF, SSNO. Modellierung von Flut Katastrophenmanagement

13	Fire ontology network http://www.semsorgrid4env.eu/index.php/ontologies/12-fire-ontology-network	Integriert SWEET. Modellierung von Feuer Katastrophenmanagement
14	Crisis Identification Ontology / Crisis Response Ontology	Humanitäre Bedürfnisse und Maßnahmen mit Beziehungen zu Auswahlkriterien
15	Solid Waste Management Operations [Dokas, 2007]	Beschreibung von Wiederverwertungsanlagen.
16	Flood Control Domain [Mongula, 2009]	Management von Flutkatastrophen (Akteure, etc.)
17	Tsunami Ontologie [Ramar, 2012]	Spezielle Tsunami Sensorik, Warnmeldungen

Die *DOLCE + DnS Ultralite Ontology* (DUL) ist eine Basisontologie, die grundlegende Konzepte wie Personen, Organisationen und Rollen mit ihren Beziehungen enthält. Eine andere verbreitete Basisontologie in Bezug auf die Modellierung von Personen und deren Beziehungen ist die *Friend of a Friend* Ontologie (FOAF), die sich auf Personen und ihre Beziehungen konzentriert.

SWEET ist ein Satz von Ontologien aus der Domäne der Geo-Wissenschaften und wurde von der NASA entwickelt. SWEET Ontologien basieren auf formaler Beschreibungslogik (DL) und sind in OWL implementiert. Das Hauptziel der Ontologien ist die Verbesserung des Auffindens und der Nutzung von Geodaten.

Die GeoNames Ontologie baut auf dem *Basic Geo Vocabulary* Wortschatz auf. Dieser enthält die Darstellung von Breite, Länge und Höheninformationen im WGS84 Koordinatensystem und wurde von der W3C Semantic Web Interest Group erstellt. GeoNames modelliert nicht nur

die Struktur von Standorten und die Beziehungen zwischen ihnen, sondern bietet auch eine große Menge an tatsächlichen Daten.

Aspekte der Zeitdarstellung sind in der *Ontology of Time for the Semantic Web* (OWL-Time) modelliert. Sie bietet ein Vokabular für Dauer, Zeitintervalle und Zeitmomente.

Die *Sensor and Sensor Network ontology* (SSNO) [Compton, 2012] und *Sensor Metadata Ontology* (SMO) [Xiao, 2016] modellieren die Beschreibung von Sensorik, die SSNO zusätzlich auch deren Messwerte.

Abbildung 66 zeigt die Hauptkonzepte der SSNO. Die aufgeführten Begriffe wurden in Abschnitt 5.4.4 bereits näher beschrieben.

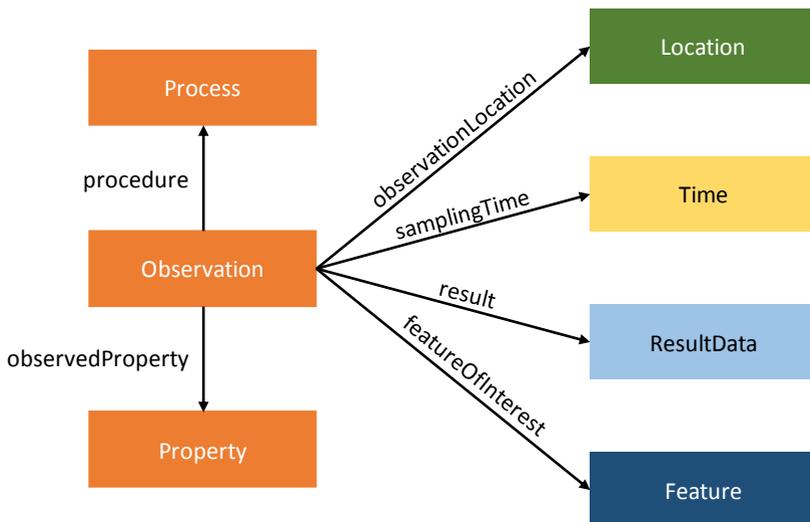


Abbildung 66: Hauptkonzepte SSNO und ihre Beziehungen

Event Model F basiert auf der Ontologie DUL und enthält zusätzliche Unterstützung für die Darstellung von Raum, Zeit, Objekten, Personen und

Beziehungen zwischen den Ereignissen. Im Gegensatz zu anderen Ontologien, die Ereignisse definieren, ermöglicht sie die Modellierung von Kausalitätsbeziehungen (Ursache/Wirkung) und die verschiedenen Interpretationen des gleichen Ereignisses.

Ontologien aus den Domänen der Frühwarnung und des Katastrophenmanagements gibt es nur wenige. Diese teilen die Eigenschaften, dass sie in wissenschaftlichen Projekten entstanden sind und meist nach Projektende nicht mehr gepflegt werden (vgl. Ontologie 12-17 in Tabelle 39). Weiterhin sind sie nur in wissenschaftlichen Publikationen beschrieben und nicht online verfügbar. Liu, Brewster und Shaw teilen diese Einschätzung in ihrem Fazit zur Untersuchung zum Stand der Forschung in diesem Bereich [Liu, 2013].

6.4.2 Modellierung der Domäne

Als Basis für die Modellierung der Domäne in RA-FWS^{NG} wird die SSNO verwendet, da sie bereits alle relevanten Basiskonzepte und Eigenschaften für die Frühwarnung enthält (s. [Moßgraber, 2015]).

Zur Erweiterung der SSNO mit domänenspezifischen Vokabularen wird die Methodik von Uschold empfohlen [Uschold, 1995]. Diese Methodik sieht folgende Schritte vor:

1. *Identifizieren des Zwecks der Ontologie*: Es ist wichtig, sich darüber im Klaren zu sein, warum die Ontologie entwickelt wird und was ihr beabsichtigter Verwendungszweck ist. Darüber hinaus müssen die Akteure (engl. Stakeholder) und ihre Umgebung definiert werden.
2. Die *Implementierung der Ontologie* teilt sich in mehrere Teilschritte auf:
 - a. Erfassung der Ontologie: Identifizierung der wichtigsten Konzepte und Beziehungen der Domäne. Erstellung von präzisen und eindeutigen Textdefinitionen für die Konzepte und Beziehungen. Identifizierung der Begriffe, die für die Konzepte und Beziehungen verwendet werden.

- b. Codierung der Ontologie: Explizite Darstellung der Konzeptualisierung in einer formalen Sprache.
 - c. Integration bestehender Ontologien: Verwendung bestehender Ontologien, die bereits im Einsatz, weit verbreitet und in der Forschungsgemeinschaft akzeptiert sind.
3. *Bewertung*: Technische Beurteilung der Ontologien in Bezug auf die gestellte Anforderungsspezifikation inkl. Testanfragen und ihr Praxiseinsatz in den zugehörigen Softwareumgebungen.
4. *Dokumentation*: Beschreibung und Erklärung aller Elemente der Ontologie, da unzureichende Dokumentation eines der Haupthindernisse für die effektive Nutzung ist.

6.4.3 Integration in SOA und EDA

Werden Ontologien eingesetzt, müssen diese auch gespeichert und verwaltet werden können. Sie werden in der Regel in einem so genannten *Triple Store* gespeichert. Typischerweise stützen sich Informationssysteme, die eine Wissensbasis nutzen, auf etablierte und effiziente Triple-Store-Speicher. Beispiele für beliebte Triple-Stores sind OWLIM [Bishop, 2011], Sesame [Broekstra, 2002], Jena [Caroll, 2004] und Virtuoso [Erling, 2009]. Obwohl diese große Mengen an gespeicherten Daten verarbeiten können (einige Billionen Triple), beschränken sich ihre Reasoning Fähigkeiten in der Regel auf RDF/RDF (S) oder die effizient handhabbaren Varianten von OWL 2 (zum Beispiel OWL 2 RL oder QL). Heutige Systeme haben solche Wissensbasen fest integriert und sprechen diese über eine komplexe APIs wie zum Beispiel Jena [Caroll, 2004] oder die OWL API [Horridge, 2008] an. Allerdings versagt diese Lösung, wenn das Informationssystem auf einer SOA oder EDA basiert und die Wissensbasis ebenfalls als Dienst implementiert werden muss, da obige APIs normalerweise keine Netzwerk-Protokolle oder Dienst-schnittstellen unterstützen. Abbildung 67 verdeutlicht das Problem.

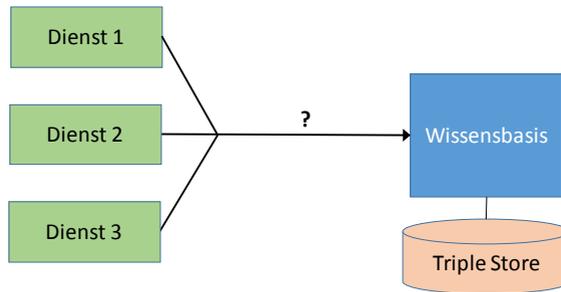


Abbildung 67: Wie greifen Dienste in einer verteilten Umgebung auf die Wissensbasis zu?

Ideen für die Verwendung von Ontologien in einer SOA wurden in den europäischen Projekten ORCHESTRA [Usländer, 2007] und SANY (s. Abschnitt 3.2.3) entwickelt und umgesetzt. Im Rahmen der beiden Projekte wurde ein Katalog für Ontologien benötigt. Der Katalog stellte über seine Dienstschnittstelle Operationen zur Verfügung, um vollständige Ontologien (TBox und ABox jedoch separat) zu lesen, zu schreiben und zu löschen. D.h. in der SOA konnten nur vollständige Ontologien zwischen den Diensten ausgetauscht werden und keine Datensätze. Zur Lösung des Problems wurden die zwei folgenden Punkte untersucht und getestet [Moßgraber, 2012b]:

1. Eine Schnittstelle zur Wissensbasis, die Funktionen einer Dienstschnittstelle mit der einer API kombiniert.
2. Eine sitzungsbasierte Instanziierung von Ontologien. D.h. zur Optimierung wird eine Wissensbasis für einen Auftrag (eine Anfrage, eine Benutzersitzung, etc.) auf der Ebene des Informationssystems betrachtet, instanziiert. Diese Wissensbasis enthält nur die instanziierten Inhalte, die relevant für den aktuellen Auftrag sind. Dies ermöglicht die Größe der Wissensbasis klein zu halten, sodass OWL 2 DL Reasoning in akzeptabler Zeit auszuführen ist.

6.4.3.1 Schnittstelle zur Wissensbasis

Als direkten Ansatz könnte man versuchen, eine Dienstschnittstelle zu entwerfen, die exakt einer existierenden API, wie zum Beispiel der OWL-API entspricht. Es gibt jedoch mehrere Nachteile bei einem solchen Versuch.

1. Die verfügbaren APIs sind meist objektorientiert entworfen. Das bedeutet, dass ein Methodenaufruf meist ein weiteres Objekt zurückliefert, auf dem weitere Methoden aufgerufen werden. Im Gegensatz dazu sind in der Regel Dienstschnittstellen nicht objektorientiert und verfügen über eine *flache* Reihe von Operationen. Die Funktionalität könnte mit komplexen Datenstrukturen nachgebildet werden, jedoch ist die Implementierung (auch auf dem Client) kompliziert, und es ergibt sich die Problematik der Datensynchronisation.
2. In einer SOA sollte die Kommunikation minimal gehalten werden, da sie zeitaufwändig ist und das Gesamtsystem verlangsamt. Bei einer objektorientierten API, wie der OWL-API, würde dann für jeden Methodenaufruf eines Objektes ein Netzwerkverkehr stattfinden.

Anstelle der Verwendung einer API besteht die Möglichkeit, Abfragesprachen wie SPARQL zu verwenden. Mit der Erweiterung SPARQL Update können die meisten API-Funktionalitäten abgedeckt werden. Ein Nachteil ist jedoch, dass Modifikationen an der Ontologie, die über eine API recht einfach sind, dazu neigen, komplex zu sein, wenn sie in SPARQL formuliert werden. SPARQL hat weiterhin den Vorteil, dass die Spezifikation die Netzwerkfähigkeit vorsieht (Remote SPARQL).

Das Ziel ist nun, die Vorteile der beiden Ansätze zu maximieren, d.h. es muss ein Kompromiss zwischen Komfort, Mächtigkeit und Performanz gefunden werden. Dabei soll die folgende Funktionalität vom Dienst der Wissensbasis (KBAS, engl. Knowledge Base Access Service) unterstützt werden.

- Speicherung, Aktualisierung oder Löschen verfügbarer Ontologien
- Abrufen von Teilen oder vollständiger gespeicherter Ontologien
- Abfrage der Ontologie Schemata, wie die Liste der Begriffe und der unterstützten Eigenschaften eines bestimmten Konzepts
- Einfügen von Ist-Daten in die Wissensbasis
- Abfragen einer oder mehrere Ontologien

Als Vorgehensweise, um eine gute Balance zwischen feingranularer API-Funktionalität und einer sehr groben zu finden, wurde mit SPARQL (und SPARQL-Update) begonnen. Danach wurden Operationen, die in einer SPARQL-Abfrage sehr komplex zu formulieren sind oder eine lange Verarbeitungszeit benötigen, als separate Funktion optimiert.

Die wichtigsten Operationen der KBAS Schnittstelle sind in Tabelle 40 dargestellt.

Tabelle 40: Schnittstelle zum Knowledge Base Access Service (KBAS)

queryOntology	Sendet eine Anfrage an eine der im KBAS gespeicherten Ontologien. Die Abfrage muss in SPARQL formuliert sein.
setOntology	Erzeugt eine neue Session-Ontologie-Instanz auf einer konfigurierbaren Basisontologie. Als Rückgabewert wird ein Bezeichner geliefert, der für die weitere Arbeit mit der Session-Ontologie verwendet werden kann (s. Abschnitt 6.4.3.2).
deleteOntology	Entfernt eine vorhandene Sitzung-Ontologie-Instanz aus dem KBAS (s. Abschnitt 6.4.3.2).
addABoxStatements	Diese Funktion ermöglicht es, Daten in eine Ontologie einzufügen. Sie ist einfacher zu benutzen als SPARQL-Update.
removeABoxStatements	Diese Funktion ermöglicht Faktenwissen (Daten) aus einer Ontologie zu löschen. Sie ist einfacher zu benutzen als SPARQL-Update.

applyRules	Wendet eine beliebige Menge von Regeln auf einer Session-Ontologie-Instanz an. Die neuen Fakten werden der Ontologie automatisch hinzugefügt.
-------------------	---

6.4.3.2 Sitzungsbasierte Instanziierung

Oft ist es möglich, die benötigten Daten für die Entscheidungsunterstützung (zum Beispiel die Informationen über die aktuelle Situation) zu begrenzen. Ein typisches Beispiel ist ein Frühwarnsystem, das regelmäßig (zum Beispiel stündlich) mit neuen Daten versorgt wird. Nur einen Bruchteil der großen Menge an gesammelten Daten ist in der Regel relevant für eine einzige Analyse-Anfrage. In solchen Situationen ist es sinnvoll, aus dem Datenspeicher nur die Daten zu extrahieren, die relevant sind und damit eine kleine Session-Ontologie zu instanziierten, auf der ein effektives Reasoning ausgeführt werden kann. Ein weiterer Vorteil der sitzungsbasierten Ontologieinstanziierung ist, dass diese aufgrund ihrer geringen Größe im Hauptspeicher gehalten werden kann, wodurch eine weitere Verbesserung der Leistung erreicht wird.

Viele Reasoner sind nicht sicher für parallele Anfragen (beispielsweise Hermit² und Pellet³). Dadurch ist eine synchronisierte Behandlung von Anfragen notwendig - mit einem großen Nachteil für die Leistung. Dies wird durch Session-Ontologien vermieden.

Ein offensichtlicher Nachteil dagegen ist, dass die Ontologie zunächst instanziiert und mit Daten befüllt werden muss. Allerdings kann dieses Problem durch den Einsatz eines Pooling-Systems reduziert werden. In einem Pool wird eine Menge an instanziierten Ontologien mit Basisdaten vorgehalten. Sobald eine neue Sitzung gestartet wird, wird die erste freie Ontologie dieser Sitzung zugeordnet. Da die Ontologieinstanz mit genau einer Benutzer-Session verbunden ist, wird keine Synchronisation benötigt.

² <http://hermit-reasoner.com/>, letzter Abruf am 4.9.2016

³ <https://github.com/complexible/pellet>, letzter Abruf am 4.9.2016

Nach der Beendigung der Session kann die Instanz effektiv im Hintergrund freigegeben und durch einen neuen leeren Klon ersetzt werden.

Beispielabfragen (s. [Moßgraber, 2012b]), die nur mit einer Hauptontologie arbeiteten, benötigten eine Verarbeitungszeit im Bereich von zwei bis vier Minuten. Dies ist für den Praxiseinsatz inakzeptabel. Durch die Umstellung auf die Session-Ontologien konnte die Bearbeitungszeit auf den Bereich von 10 bis 30 Sekunden reduziert werden (auf einem *einfachen* Arbeitsplatzrechner mit Intel i5 Prozessor und 4 GB Hauptspeicher). Etwa 95% dieser verbleibenden Zeit entfällt auf den Datenabruf, die Datenfusion und die Generierung des Antworttextes.

6.5 Verbreitung von Warnmeldungen

Personalisierte Warnmeldungen können mittels Ontologien und Regeln generiert werden. Ein wenig untersuchter Aspekt ist die Generierung mehrsprachiger Meldungen bei länderübergreifenden Warnsystemen. Hier wird meist nur mit statischen vorkonfigurierten Texten gearbeitet. Im EU Projekt PESCaDO (*Personalized Environmental Service Configuration and Delivery Orchestration*) wurde die Generierung mit Hilfe von Ontologien erforscht [Wanner, 2015]. Als Anwendungsfall wurde die Frühwarnung für Gesundheitsgefährdungen mit Bezug auf die aktuelle Situation einer Person untersucht. Beispielsweise wird einem Mann mit starken Allergien gegen Pollen von einer Fahrradtour abgeraten, wenn starker Pollenflug herrscht. Abbildung 68 und Abbildung 69 zeigen jeweils ein Beispiel für personalisierte Warnungen mit Bezug auf Zeit, Ort und persönlichen Interessen bzw. Gefährdungen (Asthmatiker, Allergiker).

Situation in the selected area between 27/02/2011 (00h00) and 28/02/2011 (00h00). The nitrogen dioxide warning threshold value ($400\mu\text{g}/\text{m}^3$) was exceeded between 23h00 and 00h00 ($422\mu\text{g}/\text{m}^3$) and between 22h00 and 23h00 ($451\mu\text{g}/\text{m}^3$). The temperature was 0°c , the wind weak and there was no rain. There is no data available for humidity.

Nitrogen dioxide warning: nitrogen dioxide causes respiratory symptoms especially in children and asthmatics, because high concentrations of this gas cause contraction of the bronchial airways. It may increase the sensitivity of the airways to other irritants such as cold air and pollen.

Abbildung 68: Meldung für Person, mit Interesse an Feinstaubbelastung in der Innenstadt

Situation in the selected area between 01/01/2012 (00h00) and 01/01/2012 (20h00). The air quality will be very poor, with the very high concentration of fine particles. The concentration of thoracic particles will be high. There is no data available for sulfur dioxide, uv, rain, wind and temperature.

Air quality index recommendation: adverse health effects are possible on sensitive subpopulation.

Fine particles recommendation: avoid excessive physical activity if you experience symptoms.

Thoracic particles recommendation: avoid excessive physical activity if you experience symptoms.

Abbildung 69: Meldung für Person, die eine Wanderung unternehmen möchte

Die Meldung kann auf Schwedisch, Finnisch und Englisch generiert werden [Bouayad-Agha, 2012]. Abbildung 70 enthält dazu ein Beispiel.

<p>Situation in the selected area between 07/05/2012 (00h00) and 08/05/2012 (17h00). <i>The birch pollen count will be abundant. The air quality will be very poor, with the very high nitrogen dioxide and very high fine particles concentrations. The minimum temperature will be -1°C and the maximum temperature 9°C, the wind will be weak, the rain light and the sky condition between clear and partly cloudy.</i> Pollen recommendation: <i>Most of the sensitive people have symptoms.</i> Air quality index recommendation: <i>Adverse health effects are possible on sensitive subpopulation.</i> Nitrogen dioxide warning: <i>Nitrogen dioxide causes respiratory symptoms especially in children and asthmatics, because high concentrations of this gas cause contraction of the bronchial airways. It may increase the sensitivity of the airways to other irritants such as cold air and pollen.</i> Fine particles recommendation: <i>Avoid excessive physical activity if you experience symptoms.</i></p> <p>Olosuhteet valitulla alueella 7.5.2012 (klo 00:00) ja 8.5.2012 (klo 17:00) vllil. <i>Ilmassa on paljon koivun siitepölyä. Ilmanlaatu on erittäin huono erittäin korkeiden tyypidioksidipitoisuuksien ja erittäin korkeiden pienhiukkaspitoisuuksien vuoksi. Alin lämpötila on -1°C ja ylin lämpötila 9°C, ja sää vaihtelee selkeästä puolipilviseen. Heikkoa tuulta. Heikkoa sadetta.</i> Siitepölysuositus: <i>Herkät ihmiset saavat oireita.</i> Ilmanlaatuindeksisuositus: <i>Terveyshaitat ovat mahdollisia herkillä väestöryhmillä.</i> Tyypidioksidivaroitus: <i>Tyypidioksidi lisää hengityselinoireita erityisesti lapsilla ja astmaattikoilla, koska se korkeina pitoisuuksina supistaa keuhkoputkia. Tyypidioksidi voi lisätä hengitysteiden herkkyyttä muille ärsykkeille, kuten kylmälle ilmalle ja siitepölyille.</i> Pienhiukkassuositus: <i>Vältä voimakasta räsitusta, mikäli saat oireita.</i></p> <p>Förhållandena på det valda området mellan 07/05/2012 (00h00) och 08/05/2012 (17h00). <i>Björkpollenhalten är hög. Luftkvaliteten är mycket dålig i och med den mycket höga kvävedioxidhalten och den mycket höga finpartikelhalten. Den lägsta temperaturen är -1°C och den högsta temperaturen 9°C och vinden är svag. Lätt regn. Klart eller svag molnighet.</i> Pollen: <i>De flesta känsliga personer kan ha allergiska symptom.</i> Index för luftkvalitet: <i>Hälsö-olägenheter är möjliga bland känsliga befolkningsgrupper.</i> Kvävedioxid: <i>Kvävedioxiden ökar andningsorgansymptomerna speciellt bland barn och astmatiker, eftersom den höga kvävedioxidhalten sammandrar luftvägar. Kvävedioxiden kan öka känsligheten för andra irriterande, till exempel för kall luft eller pollen.</i> Finpartiklar: <i>Undvik hårda ansträngningar, ifall du får symptomerna.</i></p>

Abbildung 70: Die gleiche Meldung in Englisch, Finnisch und Schwedisch

Der Ablauf der mehrsprachigen Meldungsgenerierung ist in Abbildung 71 dargestellt. Die Verarbeitungskette ist angelehnt an Reiter und Dale [Reiter, 2000], verwendet jedoch eine Wissensbasis mit Ontologien. Zur Verbesserung der Wiederverwendbarkeit wurde die Ontologie in drei Module aufgeteilt.

1. Domänen Ontologie: Diese Ontologie enthält das domänenspezifische Wissen, wie Umweltdatensätze (Temperatur, Windgeschwindigkeit, Pollenkonzentrationen, CO₂-Konzentration, etc.), Datenquellen (Wetterstationen, Luftqualitätsstationen, etc.),

geografische Informationen (Bereiche in denen Messdaten verfügbar sind, Bereich der Benutzeranfrage) und Benutzerprofile (Alter des Benutzers, Empfindlichkeit gegen Umgebungsbedingungen (Allergien, Asthma)).

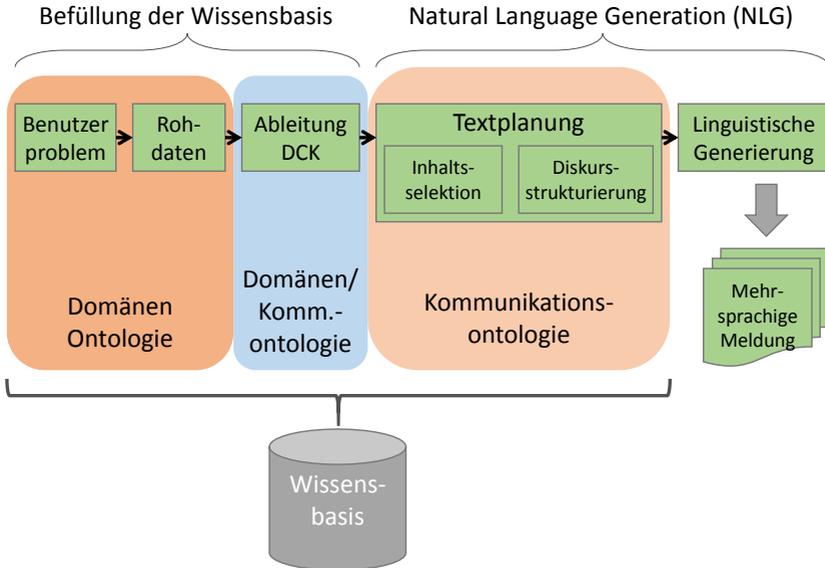


Abbildung 71: Ablauf der mehrsprachigen Meldungsgenerierung

2. Domänen-Kommunikationsontologie: Diese Ontologie enthält weitere personalisierte Inhalte, beispielsweise zur Datenaggregation (Minimum, Maximum und Mittelwert berechnet über die Zeitperiode der Anfrage), qualitativen Skalierung von numerischen Daten, auf Benutzer zugeschnittene Empfehlungen und Warnungen, die durch die Daten in der Ontologie ausgelöst wurden (zum Beispiel eine Implikation

Beziehung zwischen Pollenrate und zugehöriger Empfehlung bei einer Grenzwertüberschreitung).

3. Kommunikationsontologie: Die Kommunikationsontologie enthält die Konzepte und Beziehungen, die von den beiden Textplanungsmodulen benötigt werden: Inhaltsselektion und Diskursstrukturierung.

Die detaillierte Beschreibung des Generierungsprozesses würde an dieser Stelle zu weit führen und es sei auf [Bouayad-Agha, 2012] und [Wanner, 2015] verwiesen.

6.6 Benutzerschnittstelle

Zum besseren Situationsverständnis können Benutzer durch das Informationsnetz der Wissensbasis navigieren und semantische Anfragen stellen. Im Abschnitt 6.2.2 über den semantischen Katalog wurden dazu mehrere Beispiele gezeigt. Weitere Beispiele finden sich im Anwendungskapitel (Kapitel 1) zu RA-FWS^{NG}.

7 Automatisierung von Arbeitsabläufen

Workflows werden verwendet, um Dienste zu orchestrieren, Informationen aus verschiedenen Quellen zu sammeln und die menschliche Beteiligung an den Entscheidungsunterstützungsprozessen zu strukturieren. Natürlich kann diese Funktionalität auch mit klassischen Programmiersprachen implementiert werden, jedoch bietet die Nutzung von Workflows einige Vorteile. Hier ist vor allem die graphische Notation von Workflows zu nennen. Diese ermöglicht eine verbesserte Diskussion mit nicht-technischem Personal und ein besseres Verständnis des Prozesses, der beteiligten Akteure und die einzelnen Aufgaben des Prozesses.

Zusätzlich kann der Status einer Workflow-Instanz graphisch veranschaulicht werden, indem zum Beispiel die gerade ausgeführte Aufgabe mit zugehörigen Statusinformationen hervorgehoben wird. Dies geht Hand in Hand mit den Protokoll- und Trace-Fähigkeiten einer Workflow-Engine, die verwendet werden können, um ein Protokoll der ausgeführten Arbeitsabläufe und Aufgaben zu erstellen. Dieses Protokoll kann dann für die Fehlersuche, die *post mortem*-Analyse und die Optimierung von Prozessen verwendet werden.

7.1 Nutzung von BPMN

Das *Eidgenössische Finanzdepartement* der Schweiz hat den Einsatz von BPMN analysiert und führt folgende Vorteile an [EFD, 2011]:

1. BPMN ist ein Standard, sodass Tools zur Verfügung stehen und die Prozessdiagramme zwischen den Tools austauschbar sind,

- d. h., die Bindung an einen einzelnen Hersteller ist tendenziell kleiner geworden.
2. Empirische Untersuchungen zeigen, dass BPMN heute die Prozessmodellierungssprache mit der größten weltweiten Verbreitung ist [Patig, 2010]. BPMN-Diagramme werden also von einem großen Publikum verstanden.
 3. BPMN-Diagramme beschreiben Prozesse für die Dokumentation und Kommunikation und können gleichzeitig als Grundlage für die Automatisierung von Prozessen in einer Workflow-Engine dienen.
 4. Die BPMN-Konstrukte und die Darstellung der Prozessabläufe sind auch ohne fundierte Kenntnisse in der Prozessmodellierung leicht verständlich; dies zeigten alle Interviews und Zwischenpräsentationen, die im Projektablauf mit den Fachvertretern durchgeführt wurden.

Diese aufgeführten Punkte bestätigten sich auch beim Einsatz für Frühwarnsysteme mit RA-FWS^{NG} in den Projekten TRIDEC [Häner, 2013] und Safewater [Santamaria, 2015].

7.2 Das Workflow-Management-System

7.2.1 Architektur

Abbildung 72 zeigt die Architektur des Workflow-Management-Systems. Den Kern stellt die *Workflow-Engine* dar. Sie ist verantwortlich für die Ausführung der *Prozesse*. Prozesse sind Instanzen von Workflows, die in einem Archiv gespeichert sind. Zur Anbindung an die MOM zur Kommunikation dient der *Message-Handler*. Ein Arbeitsschritt im Prozess wird als *Task* bezeichnet. Spezifische Tasks können über eine Programmierschnittstelle implementiert werden.

Workflows neigen dazu, schnell unübersichtlich zu werden, wenn komplexe Entscheidungskriterien modelliert werden müssen [Rücker, 2016]. Dies ist

gerade bei Frühwarnsystemen regelmäßig der Fall. Als Lösung dieses Problems empfiehlt sich die Integration von Entscheidungstabellen, um regelbasierte adaptive Workflows zu realisieren. Entscheidungstabellen sind ein erprobtes Werkzeug und werden zum Beispiel intensiv im Finanzwesen eingesetzt, um komplexe Regelsätze durch einen Fachexperten modellieren zu lassen.

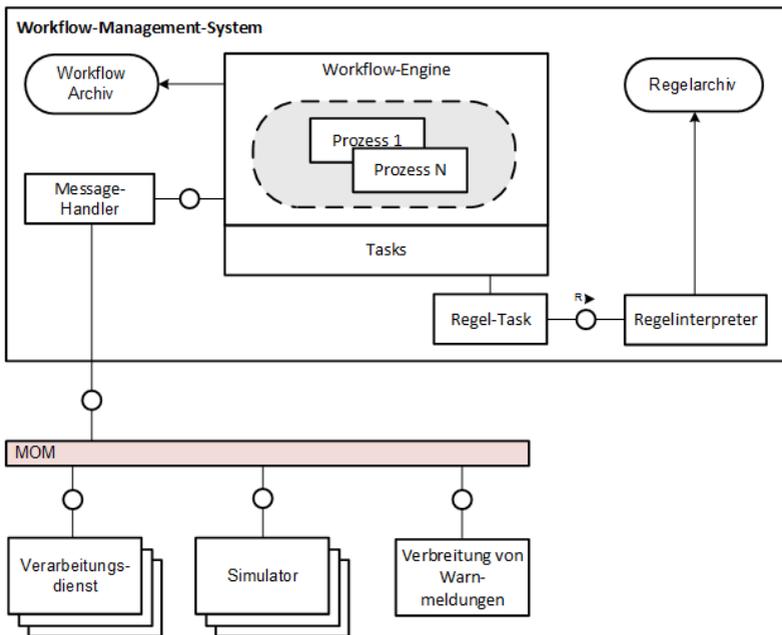


Abbildung 72: Aufbau des Workflow-Management-Systems mit integriertem Regelinterpreter

Im Workflow taucht dann nur ein verständlich benannter Entscheidungspunkt auf, der intern von einem Regelinterpreter (engl. Rule-Engine) abgearbeitet wird. Die zu überprüfenden Regeln sind in einem Regelarchiv gespeichert. Zum Zugriff auf den Regelinterpreter aus einem

Workflow dient der *Regel-Task*. Abbildung 73 zeigt den Ablauf der Interpretation eines Regelsatzes aus einem Prozess heraus.

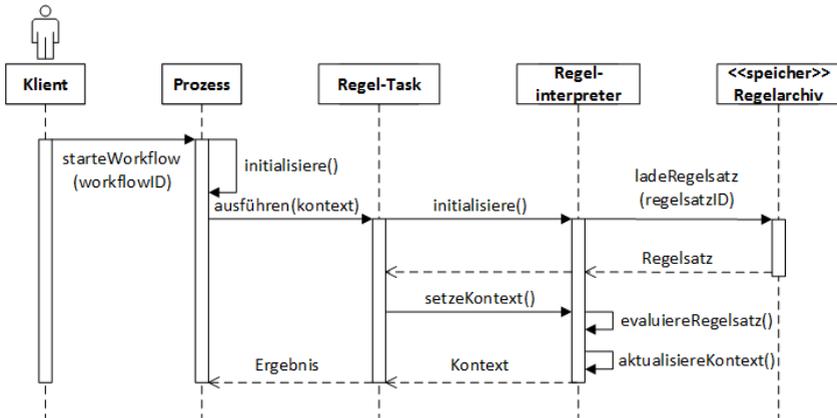


Abbildung 73: Ablauf der Integration des Regelinterpreters

7.2.2 Implementierung

*Activiti*¹ ist eine leichtgewichtige Workflow- und BPM-Plattform. Ihr Kern ist eine Prozess-Engine für Java, die BPMN in der Version 2 unterstützt. Activiti ist Open-Source und wird unter der Apache-Lizenz vertrieben. Activiti kann in jede Java-Anwendung integriert oder auf einem Server, einem Cluster oder in der Cloud ausgeführt werden. Activiti verwendet eine sogenannte *Prozess Virtual Machine* (PVM), die als Mikrokern für Workflow-Prozesse angesehen werden kann. BPMN 2 Workflows werden analysiert und in Code übersetzt, der von der PVM ausgeführt werden kann.

¹ <http://activiti.org/>, letzter Abruf am 4.9.2016

*Drools*² ist eine Business Rules Management System (BRMS) Lösung. Sie stellt eine Business Rules Engine (BRE) und eine webbasierte Anwendung zur Erstellung und dem Management von Regeln (*Drools Workbench*) bereit.

Eine Integration von Drools wird von Activiti nicht direkt unterstützt, weshalb ein spezifischer Regel-Task (s.o.) erstellt werden musste.

7.3 Aufruf von Komponenten

Ein Prozess muss in einem Arbeitsschritt Zugriff auf Datenquellen oder Dienste des Systems haben, um seine Arbeit erledigen zu können. Zur Anbindung bietet Activiti eine Erweiterungsmöglichkeit über sogenannte *Task Templates*. Abbildung 74 zeigt die verschiedenen Erweiterungen³, die dem Prozess zu Verfügung stehen sollten.

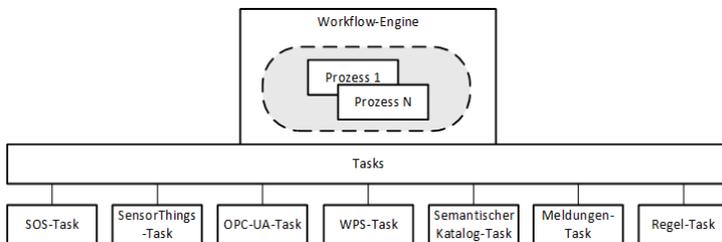


Abbildung 74: Verschiedene Hilfstasks für BPMN

Dies sind zum einen die Zugriffsmöglichkeiten auf die in Abschnitt 5.5.2 beschriebenen Datenquellen (SOS Server, SensorThings API Server, OPC-UA Server, etc.) und zum anderen Tasks zum Zugriff auf den semantischen Katalog, die Meldungsverbreitung und den Regelinterpretier.

² <http://www.drools.org/>, letzter Abruf am 4.9.2016

³ Diese sind der besseren Lesbarkeit wegen nur als *Task* und nicht als *Task Template* bezeichnet.

8 Anwendungsbeispiele

Im Folgenden wird die Anwendung von RA-FWS^{NG} anhand von zwei Beispielen gezeigt, zunächst anhand der Domäne zur Tsunami-Frühwarnung und danach am Anwendungsfall der Frühwarnung vor verunreinigtem Trinkwasser.

Da die vollständige Spezifikation beider Architekturen den Umfang dieser Arbeit bei weitem sprengen würde, werden nur die interessantesten Aspekte herausgegriffen und beschrieben. Vor jedem der beiden Anwendungsfälle befindet sich deshalb eine Tabelle, die die RA-FWS^{NG} Arbeitsschritte enthält, die vorgestellt werden.

8.1 Tsunami-Frühwarnung

Die Forschungsergebnisse, die in diesem Kapitel vorgestellt werden, wurden im Rahmen des EU Projekts TRIDEC erarbeitet [Moßgraber, 2012a], [Moßgraber, 2013].

Die Problematiken eines Tsunami-Frühwarnsystems (TFWS) wurden bereits im Abschnitt 3.4.3 deutlich. Ein TFWS ist über eine große geographische Region verteilt und besteht aus einer großen Anzahl von Computersystemen und Sensoren, die, um interagieren zu können, eine gemeinsame *Sprache* sprechen müssen (semantische Ebene). Derzeitige Systeme sind noch nicht vollständig gekoppelt und es ist eine Vielzahl von zusätzlichen Benutzeraktionen erforderlich, bis eine Warnung schließlich die Öffentlichkeit erreicht. Durch die Anbindung bestehender Systeme, die Fusionierung ihrer Informationen und die Nutzung für simulierte Prognosen kann ein besseres Verständnis der Situation erreicht werden.

Die interessantesten Aspekte der Modellierung der Systemarchitektur mit RA-FWS^{NG} sind in Tabelle 41 dargestellt (Teilmenge von Tabelle 37) und werden in den folgenden Abschnitten beschrieben.

Tabelle 41: Vorgestellte Arbeitsschritte für den Anwendungsfall Tsunami

#	S. Abschnitt	Arbeitsschritt
Unternehmenssicht		
4	5.1.4	Festlegung der Anwendungsfälle mit Hilfe von SOPs.
System von Systemen Sicht		
5	5.2.1	Prüfung ob ein SoS vorliegt.
6 (opt.)	5.2.2	Ggf. Implementierung einer skalierbaren und resilienten Kommunikationsschicht.
Systemsicht		
8	5.3.2	Integration aller notwendigen Datenquellen.
9	5.3.6	Anbindung und Implementierung aller Verarbeitungsdienste.
Informationssicht		
15	5.4.1	Erstellung der SensorML Beschreibungen für die spezifischen Sensortypen.
Technologiesicht		
18	5.5.1 - 5.5.11	Auswahl der passenden Softwarelösungen für jede Komponente.

8.1.1 Festlegung der Anwendungsfälle mit Hilfe von SOPs

Der stark vereinfachte Ablauf der semi-automatisierten Tsunami-Frühwarnung ist in Abbildung 75 dargestellt. Das seismische Teilsystem (engl. Seismic System) detektiert ein Erdbeben und sendet deshalb eine Meldung an das Warnsystem. Nicht aus jedem Erdbeben muss ein Tsunami entstehen, d.h. es ist nun zu prüfen, ob eine potentielle Gefahr besteht.

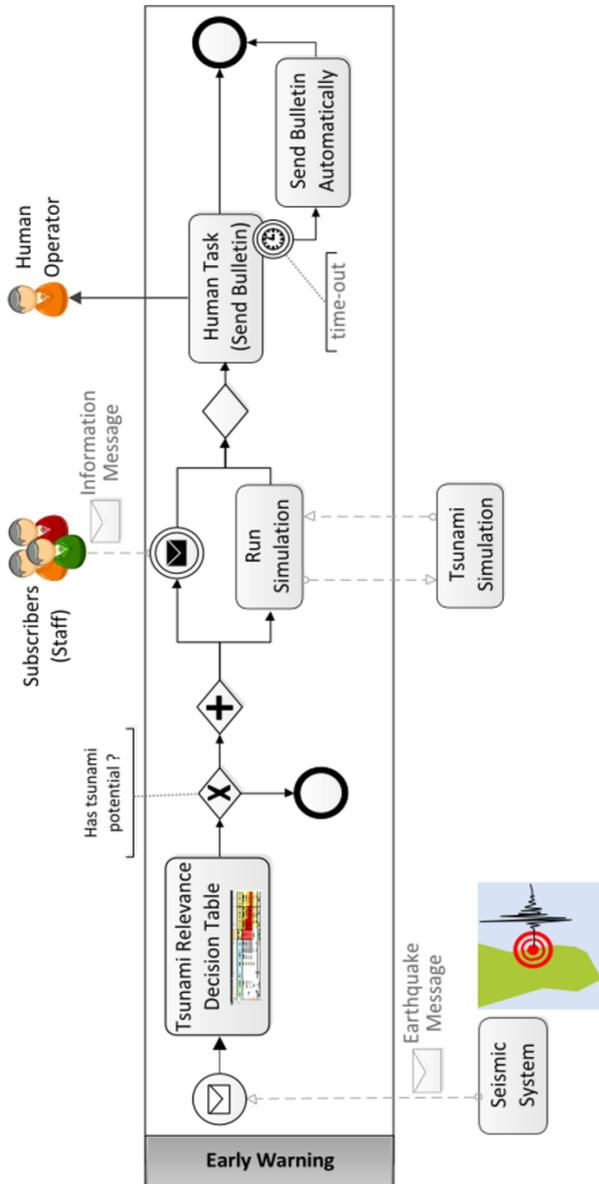


Abbildung 75: Ablauf der semi-automatisierten Tsunami-Frühwarnung

Zur ersten Prüfung wird eine Entscheidungstabelle (s. Abschnitt 7.2.1) genutzt, die abhängig von der Position und Tiefe des Bebens ermittelt, ob ein Tsunamipotential besteht oder nicht. Besteht eine Gefahr, werden parallel relevante Personen benachrichtigt und automatisch Simulationen über mögliche Ausbreitungen von Tsunamiwellen gestartet. Alle Informationen inkl. der Simulationsergebnisse werden einem Entscheider zur Verfügung gestellt, der bewertet, ob eine Warnmeldung veröffentlicht werden soll.

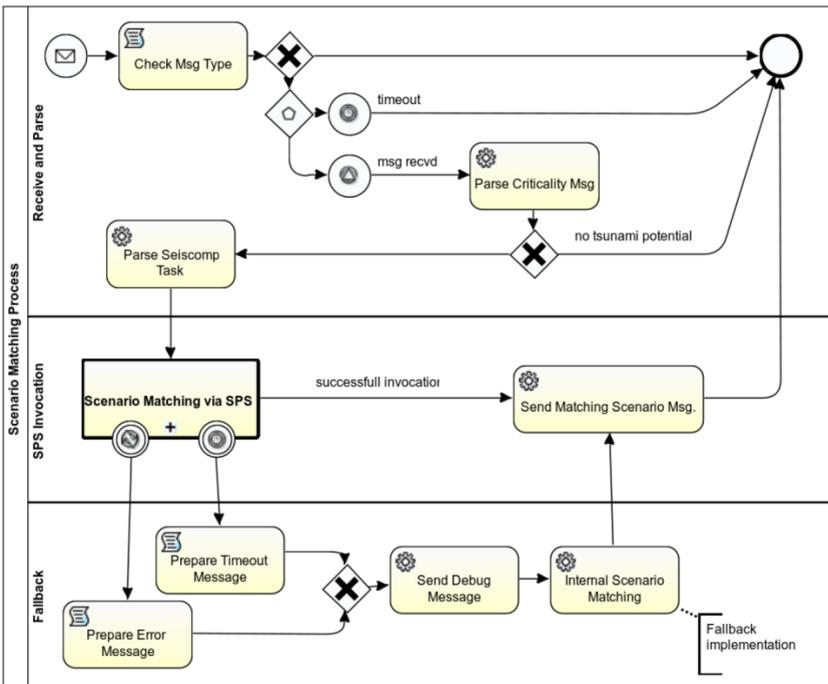


Abbildung 76: Abgleich des Szenarios mit Fehlerbehandlung

Sollte diese Person nicht verfügbar sein, wird nach einer gewissen Zeit automatisch eine Warnmeldung verschickt. Diese enthält dann einen entsprechenden Hinweis, dass sie ungeprüft versandt wurde. Dieser Detailgrad des Ablaufs eignet sich gut für die Diskussion mit Fachexperten. Für die reale Implementierung sind jedoch diverse Fehlerbehandlungen notwendig (engl. Fallback).

Abbildung 76 zeigt die Ausgestaltung für den Aufruf der Prüfung des Tsunamipotentials. Die Startmeldung könnte beispielsweise versehentlich auf dem *Topic* gelandet sein, die diesen Prozess auslöst. Diese Darstellung ähnelt nun eher schon einer Programmiersprache und ist für Endnutzer nur noch schwer nachvollziehbar.

8.1.2 Prüfung, ob ein SoS vorliegt

Ein TFWS stellt eindeutig ein SoS dar. Die zugehörige Prüfung wurde bereits in Abschnitt 5.2.1 in Tabelle 26 als Beispiel vorgestellt.

8.1.3 Implementierung einer Kommunikationsschicht

Als Lösung wurde eine MOM umgesetzt. An jedem Systemstandort wurde ein Brokercluster basierend auf Apache Qpid (s. Abschnitt 5.5.1.3) aufgesetzt. Zur Verbesserung der Skalierbarkeit und Resilienz wurden die in Abschnitt 5.2.2.1 und 5.2.2.2 gegebenen Empfehlungen, soweit in Apache Qpid vorhanden, konfiguriert bzw. teilweise nachimplementiert.

8.1.4 Integration aller notwendigen Datenquellen

Als Quelle für Sensordaten wurde auf den bereits in den GITEWS/DEWS Projekten implementierten Tsunami Sensor Bus (s. Abschnitt 3.4.3.1) zugegriffen. Da dieser die OGC SWE Konzepte und Empfehlungen umsetzt, ist damit bereits die Datenintegration auf syntaktischer Ebene abgedeckt.

Zur Beantwortung semantischer Fragestellungen und zum dynamischen Registrieren neuer Sensorik wurde eine Verbindung zum semantischen Katalog hergestellt (s. auch Abschnitt 8.1.6).

8.1.5 Verarbeitungsdienste

Die zwei wesentlichen Verarbeitungsdienste sind die Klassifikation des Tsunamipotentials und die Ausbreitungsrechnung der Tsunamiwelle. Bei der Ausbreitungsrechnung ist ein weiterer Verarbeitungsdienst nachgeschaltet, der aus den berechneten alternativen Ausbreitungsmöglichkeiten ein Ranking mit Wahrscheinlichkeiten erstellt.

Zusätzlich wurden Experimente zur Tsunamierkennung mithilfe sozialer Medien durchgeführt. Dieser Forschung liegt die Erkenntnis zugrunde, dass, wenn eine Tsunamiwelle an eine Küste trifft, die betroffenen Menschen beginnen, Meldungen in sozialen Medien zu schreiben (beispielsweise Twitter). Diese Meldungen werden durch Glasfasernetze schneller übertragen als die Ausbreitung der Tsunamiwelle. Die Bevölkerung von Küsten, die noch nicht von der Welle getroffen wurden, können noch gewarnt werden, wenn man die entsprechenden Informationen aus den sozialen Medien extrahieren kann. Um dies zu ermöglichen, wurden zwei Verarbeitungsdienste erstellt, ein Dienst zur geographischen Verortung einer Meldung und ein Dienst zur Klassifikation des Textes als Bericht über einen Tsunami. Beide Dienste wurden mit Meldungen aus Twitter (*Tweets*) getestet [Zielinski, 2013].

8.1.6 Beschreibung der spezifischen Sensortypen

Die Spezifikationen von OGC SWE sind relativ komplex. Der Grund hierfür ist das äußerst breite Spektrum von Anwendungen, für die diese Standards ausgelegt sind. Auf der einen Seite ist dadurch zwar ein flexibler Einsatz von SWE möglich, auf der anderen Seite erhöht dies den Aufwand bei der Umsetzung und Anwendung deutlich. Im EU-Projekt EO2HEAVEN wurde

deshalb eine Empfehlung für einen reduzierten Parametersatz gegeben, der sich auf die am häufigsten verwendeten Elemente konzentriert [EO2HEAVEN, 2013]. Diese Ratschläge wurden in RA-FWS^{NG} ebenfalls befolgt.

Im semantischen Katalog wurden 269 Sensorstationen mit 116 beobachteten Eigenschaften gespeichert. Jede Sensorstation ist einem sogenannten Angebot (engl. Offering) zugeordnet, die den Kontext, in dem sie verwendet wird, beschreibt, zum Beispiel Doos für Deep Ocean Observation System.

8.1.7 Softwarelösungen für jede Komponente

Die eingesetzten Technologien für RA-FWS^{NG} sind in Abbildung 77 dargestellt. Zur Umsetzung der MOM wurde Apache Qpid ausgewählt (s. Abschnitt 5.5.1.3). Als Datenquelle dient der Tsunami Sensor Bus (s. Abschnitt 8.1.4). Die Daten werden in MongoDB¹ gehalten. Als Wissensbasis wurde OWLIM [Bishop, 2011] mit der SSNO eingesetzt. Der semantische Katalog wurde mit WebGenesis[®] realisiert (s. Abschnitt 6.2.2). Als BPMS kam Activiti (s. Abschnitt 7.2.2) zum Einsatz. Als Benutzerschnittstelle wurde das aus GITEWS stammende *Command and Control User Interface* (CCUI) verwendet und die neuen Dienste integriert.

¹ <https://www.mongodb.com>, letzter Abruf am 4.9.2016

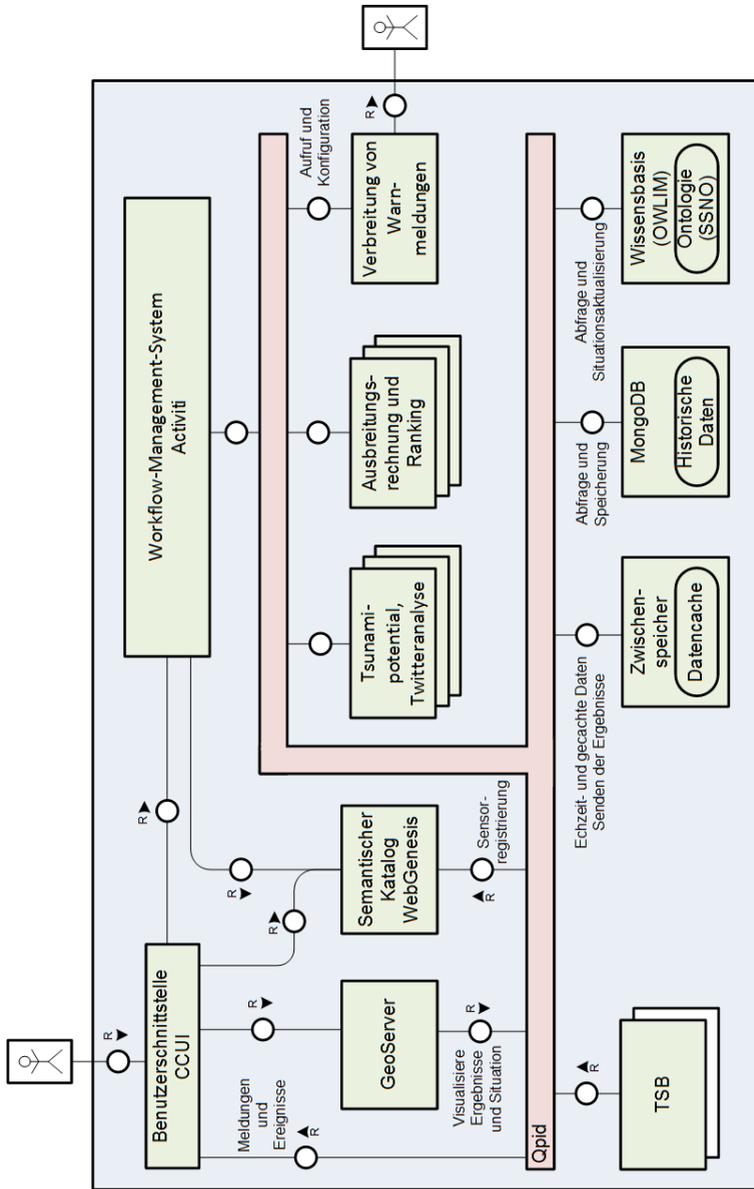


Abbildung 77: Eingesetzte Technologien für RA-FWS^{NG} in TRIDEC

8.2 Sicheres Trinkwasser

Die Forschungsergebnisse, die in diesem Kapitel vorgestellt werden, wurden im Rahmen des EU Projekts SAFEWATER erarbeitet [Bernard, 2015a].

Die Sicherheit von Trinkwasser wird zunehmend als eine große Herausforderung für die Kommunen und Wasserversorger erkannt. Diese kann durch Naturkatastrophen, Unfälle oder böswillige Angriffe bedroht werden. Im Falle einer Kontamination des Wassers verbreitet sich dieses schnell und damit weitgehend, bevor das Problem detektiert wird. Verunreinigtes Trinkwasser kann Epidemien auslösen, das Wirtschaftsleben stören und eine Massenpanik verursachen. Die erste Generation von Softwarelösungen und Sensoren für die Verwaltung der Trinkwassersicherheit, wie *Guardian Blue* der Firma Hach² oder *Canary* der US Environmental Protection Agency (EPA)³, ermöglichen bereits die systematische Erfassung und Auswertung von Informationen.

Diese Werkzeuge leiden noch an einer Reihe von Mängeln [Bernard, 2015b]:

1. Eine Echtzeit-Erkennung und Alarmfunktionen sind nicht vorhanden oder nicht ausreichend.
2. Die jeweiligen Einschränkungen der Ausbreitungsmodelle machen die effektive Lagebeurteilung von potenziell kontaminierten Zonen sehr schwierig.
3. Bisher existiert kein generischer Ansatz für die Online-Kalibrierung des Hydraulik- und Transportmodells.
4. Eine integrierte Unterstützung des vollständigen Ablaufs vom Auftritt eines Ereignisses, der Reaktionen darauf, bis zum Schließen des Ereignisses ist nicht vorhanden.

² <http://www.hach.com/>, letzter Abruf am 4.9.2016

³ <https://software.sandia.gov/trac/canary>, letzter Abruf am 4.9.2016

5. Die Verfügbarkeit von Online-Sensoren, die verwendet werden können, um Verschmutzungen und Bedrohungen der Trinkwasserqualität zu erfassen, ist noch sehr begrenzt.
6. Simulatoren sind nicht in den Arbeitsablauf eingebunden und müssen separat bedient werden.

Zur Adressierung dieser Punkte mit Schwerpunkt auf 1, 4 und 6 wurde das Frühwarnsystem *Event Management System (EMS)* entworfen und umgesetzt [Santamaria, 2015]. Die interessantesten Aspekte der Modellierung der Systemarchitektur mit RA-FWS^{NG} sind in Tabelle 42 dargestellt (Teilmenge von Tabelle 37) und werden in den folgenden Abschnitten beschrieben.

Tabelle 42: Vorgestellte Schritte für Sicheres Trinkwasser

#	S. Abschnitt	Arbeitsschritt
Unternehmenssicht		
4	5.1.4	Festlegung der Anwendungsfälle mit Hilfe von SOPs.
Systemsicht		
8	5.3.2	Integration aller notwendigen Datenquellen.
9	5.3.6	Anbindung und Implementierung aller Verarbeitungsdienste.
10	5.3.6.4	Prüfung, ob ein BigData Problem vorliegt.
12	5.3.7	Anbindung und Implementierung aller Simulationen.
13	5.3.11	Identifikation und Integration des benötigten Kartenmaterials.
14	5.3.12	Spezifikation der Benutzeroberfläche.
Technologiesicht		
18	5.5.1 - 5.5.11	Auswahl der passenden Softwarelösungen für jede Komponente.

8.2.1 Festlegung der Anwendungsfälle

Im Trinkwasserbereich wird statt des Begriffs Frühwarnung häufig das Wort Ereignismanagement (engl. Event Management) verwendet [Dąbrowski, 2015]. Abbildung 78 zeigt den Ablauf des Ereignismanagements, der sich im Wesentlichen mit dem des Katastrophenmanagements deckt.



Abbildung 78: Einfacher Ablauf des Ereignismanagements

Wenn eine abweichende Situation erkannt wird, wird ein Alarm gesendet. Da es sich auch um einen Fehlalarm handeln kann, muss durch einen Entscheider ein offizielles Ereignis deklariert werden, erst dann läuft die Abarbeitung der Reaktionsaufgaben an. Wurden alle Aufgaben durchgeführt und der Normalzustand konnte wieder hergestellt werden, wird das Ereignis geschlossen.

8.2.1.1 Deklariere Ereignis

Der BPMN Prozess zur Deklaration eines Ereignisses ist in Abbildung 79 dargestellt und wird in folgender Liste beschrieben.

1. Das *Event Detection System* (EDS) erkennt Änderungen in den überwachten Parametern, die von einem Kontaminierungsereignis stammen können.
2. Das EDS sendet einen Alarm an das EMS.
3. Das EMS empfängt und speichert den Alarm.
4. Das EMS zeigt den Alarm auf der Benutzerschnittstelle an.
5. Der Alarm wird bewertet und ggf. vom zuständigen Fachexperten bestätigt. Nach der Bestätigung eines kritischen Alarms werden die

folgenden Arbeitsschritte in der Liste ausgeführt. Die Evaluierung kann durch einen Regelinterpreter unterstützt werden.

6. Automatischer Versand von E-Mails an den Mitarbeiterstab.
7. Automatischer Start der folgenden Simulationen:
 - a. Simulation der Ausbreitung der Verunreinigung im Wassernetzwerk.
 - b. Berechnung des Ortes, an dem die Verunreinigung stattgefunden hat anhand der Orte der Sensoren, die den Alarm gemeldet haben.

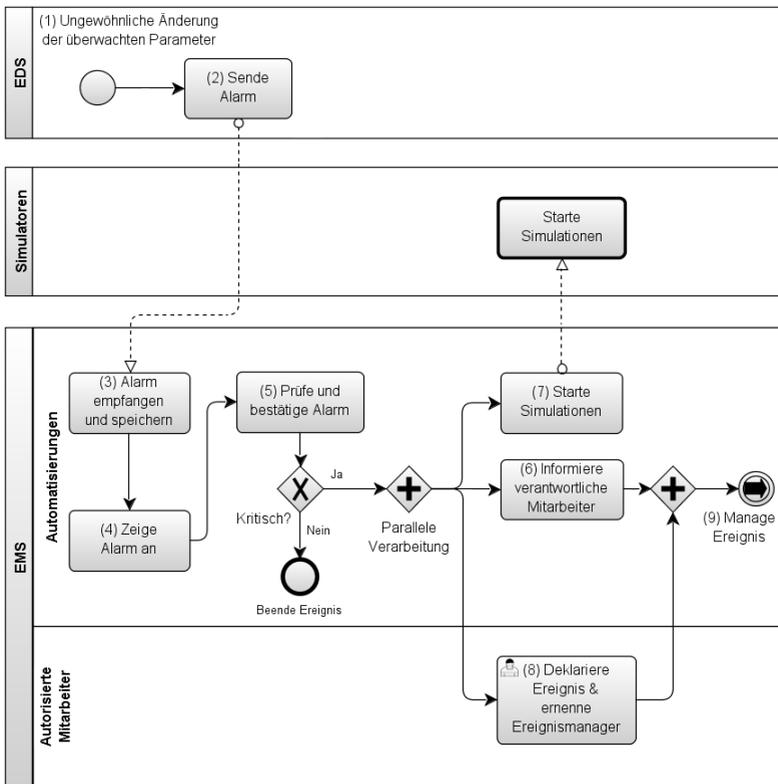


Abbildung 79: BPMN Prozess zur Deklaration eines Ereignisses

8. Ein autorisierter Mitarbeiter des Wasserversorgers erstellt ein Kontaminierungsereignis im EMS.

9. Das EMS startet die Abarbeitung der Reaktionsphase.

8.2.1.2 Reaktionsphase

Eine graphische Darstellung des Arbeitsablaufs einschließlich der direkten und fortlaufenden Aufgaben sind in Abbildung 80 dargestellt.

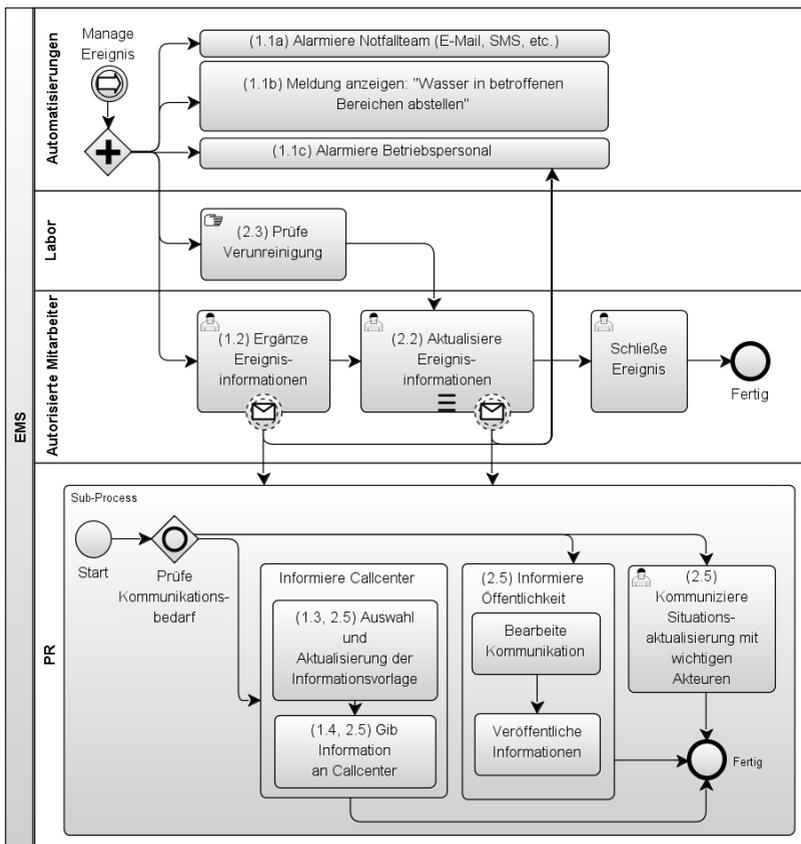


Abbildung 80: BPMN Prozess zur Reaktionsphase

Dieser Ablauf ist als grobe Vorlage zu sehen und unterscheidet sich je nach Wasserversorger und lokaler Gesetzgebung erheblich. Durch Einsatz eines BPMS ist sichergestellt, dass dieser Prozess flexibel angepasst und erweitert werden kann.

Direkte Reaktionen

Folgende Punkte können direkt nach der Erkennung eines Ereignisses ausgeführt werden.

- 1.1 Wenn das Ereignis deklariert wird, führt das EMS die folgenden Aktionen aus.
 - a) Senden von Nachrichten (E-Mail, SMS, etc.) an alle Mitglieder des Notfallteams, die erste Informationen über das Ereignis enthalten.
 - b) Anzeige, dass der Wasserfluss in den betroffenen Bereichen gestoppt werden muss. Dies ist abhängig von den Simulationsergebnissen der Kontaminierungsausbreitung.
 - c) Senden automatischer Nachrichten an das Betriebspersonal mit ersten Informationen über das Ereignis und die Aufforderung, den Wasserfluss in dem betroffenen Bereich zu stoppen, um die Ausbreitung der Verunreinigung zu verhindern bzw. zu begrenzen.
- 1.2 Ein autorisierter Benutzer ergänzt die verfügbaren Informationen über das Problem. Zusätzliche Informationen sind beispielsweise der ernannte Ereignismanager, Situationsaktualisierungen, der Arbeitsplan, betroffene Bereiche und die Kommunikation mit externen Kontakten (Polizei, Gesundheitsamt, Stadtverwaltung, ...).
- 1.3 Ein Mitglied der Öffentlichkeitsarbeit (PR) wählt und füllt die entsprechende Vorlage für die Beantwortung von Verbraucheranrufen sowie für eine Pressemitteilung.
- 1.4 Die Nachrichten werden an die Öffentlichkeit und an das Call-Center geschickt.

Fortlaufende Reaktionen

Um die Entscheidungsfindung zu unterstützen, bietet das EMS eine zentrale Lage, in der alle Informationen über laufende Ereignisse gesammelt werden. Diese zentrale Informationsstelle spiegelt die aktuelle Situation und dokumentiert geplanten Aktionen. Ein wichtiger Aspekt des Systems ist, dass alle Arbeiten protokolliert werden, um die Reaktionsmaßnahmen analysieren zu können, nachdem der Notfall beendet ist.

In der Reaktionsphase müssen zahlreiche Aktionen ausgeführt werden. Viele sind einfache Aufgaben, die keine Automatisierung benötigen (oder in anderen Systemen ablaufen) und einen Prozess unnötig komplex werden lassen. Für solche Aufgaben sind einfache Checklisten, die an bestimmten Stellen des Prozesses abgehakt werden können, besser geeignet. Im EMS ist es deshalb möglich, Vorlagen für Checklisten zu erstellen, die an bestimmten Stellen des Prozesses instanziiert werden können. Diese Aktionen sind im Diagramm in Abbildung 80 nicht aufgeführt, jedoch unten beschrieben.

- 2.1 Festlegen des betroffenen Gebiets, Vorhersagen der Kontaminationsausbreitung, Empfehlen von durchzuführenden Aktionen (zum Beispiel Ventile schließen), Visualisierung von Simulationsergebnissen, etc.
- 2.2 Lagebericht & Operationsplan: Die Benutzer können die Ereignisinformationen mit einem aktualisierten Lagebericht und einem Operationsplan erweitern. Das System fordert die stündliche Aktualisierung der Berichte.
- 2.3 Sicherstellen, dass die Kontamination real ist (zum Beispiel durch Probenahmen und chemische Analysen): Die Ergebnisse dieser Aufgabe werden dem Ereignis als neue Informationen hinzugefügt.
- 2.4 Untersuchung der Ursache der Kontamination.
- 2.5 Information der Öffentlichkeit und wichtiger Akteure, wenn neue Informationen verfügbar sind.

2.6 Erneute Wasserqualitätsprüfung, ob die Isolierung der kontaminierten Zone gelungen ist.

2.7 Bereitstellung einer alternativen Wasserversorgung.

8.2.2 Integration aller notwendigen Datenquellen

Wasserversorger setzen SCADA Systeme zur Anbindung der Sensorik ein. Das jeweilige System stellt damit bereits eine zentrale Datensammelstelle bereit, auf die zugegriffen werden kann. Vielerorts werden leider noch proprietäre Systeme eingesetzt, die keine offene Schnittstelle zur Datenabfrage unterstützen (OPC-UA verbreitet sich jedoch zunehmend in diesem Bereich). Ist dies der Fall, muss direkt auf die Datenbank des SCADA Systems zugegriffen werden und ein spezielles *Task Template* (s. Abschnitt 7.3) für den Datenzugriff erstellt werden.

8.2.3 Verarbeitungsdienste

Im System wird nur ein Verarbeitungsdienst zur Überwachung der Frühwarnparameter eingesetzt. Dies ist das *Event Detection System* (EDS) der israelischen Firma *Decision Makers Ltd*⁴. Das EDS erkennt bislang unbekannte Konstellationen von Werten der Wasserqualitätsparametern auf Basis maschinellen Lernens. Diese Ereignisse können ein Hinweis auf eine Verunreinigung des Trinkwassernetzes sein.

In vielen Fällen ist die Erkennung von Verunreinigungen schwierig, da das System etwas finden soll, das in der Vergangenheit noch nicht aufgetreten ist. Eine solche Herausforderung kann durch die Verwendung von einer Methodik des unüberwachten Lernens (engl. *Unsupervised learning*) adressiert werden. Das Lernverfahren beobachtet das normale Verhalten der Parameter und kann dadurch erkennen, wenn diese davon abweichen.

⁴ <http://www.decisionmakers.biz/>, letzter Abruf am 4.9.2016

Das EDS bietet mehrere Methoden, um einen Alarm zu erzeugen:

- Verletzung der Grenzwerte einzelner Variablen
- Auftreten seltener Kombinationen
- Ähnlichkeit der Werte zu Problemsituationen der Vergangenheit
- Verletzung von Regeln

Für diese Methoden gibt es jeweils Detektoren, die auch kombiniert werden können. Ein Detektor ist ein Algorithmus, der trainiert wird, um die spezifischen Abweichungen in den Parameterwerten zu erfassen. Ein Beispiel für einen Detektor ist ein Algorithmus zur Erkennung, ob der Wert einer Variablen außerhalb der statistischen Grenzen liegt. Ein solcher Detektor lernt die statistischen Grenzen der Wertebereiche einzelner Variablen und erzeugt einen Alarm, wenn die tatsächlichen Daten (nach einer Verzögerungszeit) gegen diese Grenzen verstoßen.

8.2.4 Prüfung, ob ein BigData Problem vorliegt

Die Messwerte der Sensorik werden heute bereits in einer relationalen Datenbank des SCADA Systems abgelegt, die diese problemlos verwalten kann. Selbst bei einem massiven Ausbau der (Online-)Sensorik im Netzwerk wird sich die Datenmenge nicht über den Terabyte-Bereich hinausbewegen. Es liegt also kein BigData-Problem vor.

8.2.5 Anbindung aller Simulatoren

Die verfügbaren Simulatoren werden vom BPMS über die MOM gesteuert. Da die Simulatoren sehr rechenintensiv sind, kann nur eine begrenzte Anzahl parallel ausgeführt werden (je nach Systemanforderungen kann dies natürlich über mehrere Rechner skaliert werden). Deshalb werden alle Simulatoren über einen zentralen *Message-Handler* angesprochen, der gleichzeitig die Verfügbarkeit der Simulatoren überwacht und verwaltet (s. Abbildung 81).

Für die Ausführung der Simulationen ist ein separater Prozess zuständig, der auf die Beendigung einer Berechnung wartet. Sobald die Berechnungen abgeschlossen sind, werden die Ergebnisse auf der GIS-Komponente des EMS angezeigt. Diese Simulationen laufen parallel zur Echtzeit-Simulation des Wassernetzes (*Online*), welche kontinuierlich die Hydraulikberechnungen durchführt, die als Basiswerte der anderen Simulatoren dienen.

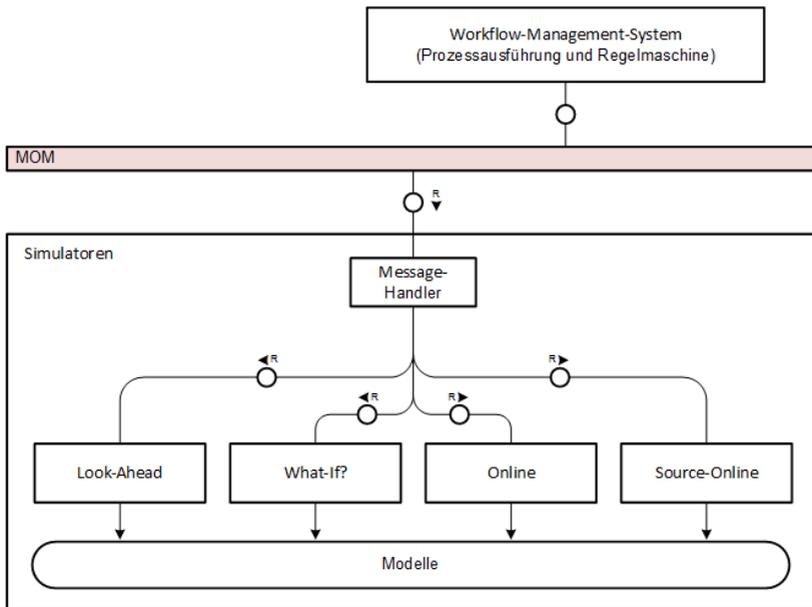


Abbildung 81: Integration von Simulatoren

Neben der Hydraulikberechnung sind folgende drei Simulatoren verfügbar:

- 1) Ausbreitungsrechnung (*Look-Ahead*)
- 2) Berechnung bestimmter Szenarien (*What-If?*), zum Beispiel was passiert, wenn bestimmte Schieber geschlossen werden
- 3) Berechnung der Quelle einer Kontamination (*Source-Online*)

8.2.6 Benötigtes Kartenmaterial

Neben einer Hintergrundkarte (Straßen, Gebäude, Flüsse, Seen, etc.) werden folgende weitere Ebenen benötigt:

- Das Rohrnetzwerk
- Positionen der Schieber (zum Schließen von Rohrabschnitten)
- Positionen der Sensorik

Neben diesen (größtenteils) statischen Daten werden als Ebenen die aufgetretenen Ereignisse und Simulationsergebnisse angezeigt. Abbildung 82 zeigt das Simulationsergebnis einer Ausbreitungsrechnung.



Abbildung 82: Simulationsergebnis einer Ausbreitungsrechnung

Die Linien stellen das Rohrleitungssystem dar. Der Kreis am oberen Rand repräsentiert den Startpunkt der Berechnung (der Ort an dem die Verunreinigung stattgefunden hat). Die Ausbreitung ist farblich im Stundentakt dargestellt. Rot ist die Ausbreitung nach einer Stunde, Orange nach zwei Stunden, gelb nach drei Stunden und grün nach vier Stunden. In der Mitte ist zusätzlich die Position eines Ereignisses mit einem Marker dargestellt.

8.2.7 Spezifikation der Benutzeroberfläche

Die Benutzeroberfläche stellt den integrierten Zugang zu allen Systemkomponenten und -funktionen bereit. Die Oberfläche ist webbasiert und passt sich automatisch an das jeweilige Endgerät an. Dadurch kann die Benutzung sowohl vom Browser eines Bürorechners als auch auf dem Mobilgerät eines Wartungsmitarbeiters erfolgen. Zwei Ausschnitte aus der Benutzeroberfläche werden im Folgenden beschrieben. Abbildung 83 zeigt die Ansicht der Alarme. Die Einfärbung der Alarme erfolgt nach ihrer Kritikalität. Es können zusätzlich manuelle Alarme eingetragen, die sich beispielsweise aus Kundenbeschwerden („Wasser ist getrübt oder riecht seltsam“) ergeben.

Alarms

Show entries Search:

Type	Severity	Title	Creation	Ack status	Latitude	Longitude
EDS	HIGH	Dom1 (2)	05.05.2016 10:39:00	✓	52,511598	13,437609
EDS	MEDIUM	Pat_Junction (8)	01.04.2016 18:19:00	🔔	0,0	0,0
EDS	INFO	Pat_Junction (8)	01.04.2016 17:00:00	🔔	0,0	0,0

Showing 1 to 3 of 3 entries

Previous **1** Next

Edit

Add

Edit

Delete

Maps

Water Network

Other

Ack Alarm

Discard Alarm

Abbildung 83: Ansicht der Alarme

Abbildung 84 stellt den Ablauf der Deklaration eines Ereignisses dar. Mehrere Alarme können einem Ereignis zugeordnet werden, d.h. diese Alarme beziehen sich auf das gleiche Problem. Nach Anwahl der Ereignisdeklaration (*Declare Event*) wird ein Dialog angezeigt, in dem ein Titel für das Ereignis eingegeben werden kann. Nach dessen Bestätigung laufen im Hintergrund bereits einige Aufgaben an (E-Mail Versand, Simulationsstart, etc.). Im erscheinenden Formular kann der Benutzer weitere Informationen zum Ereignis erfassen.

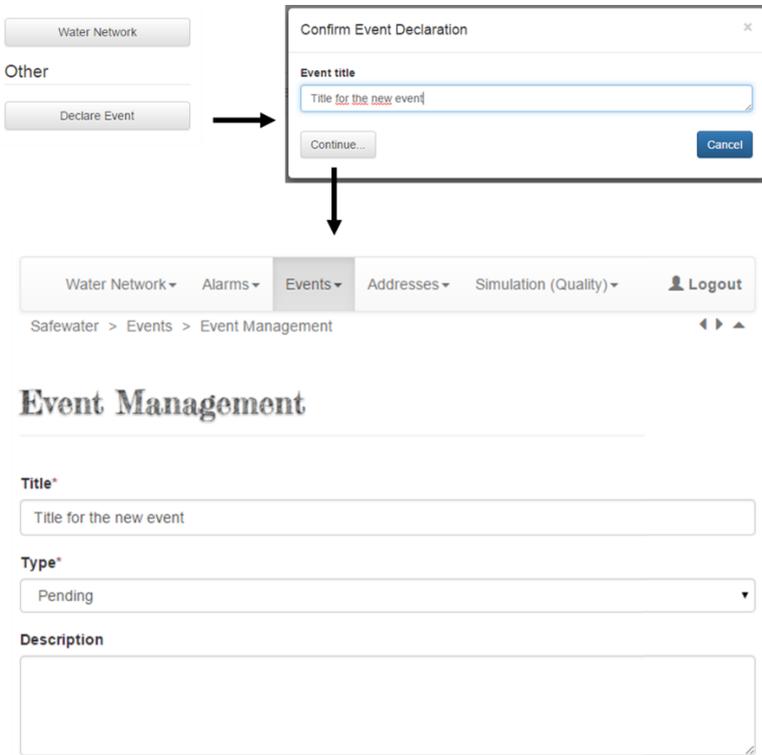


Abbildung 84: Deklaration eines Ereignisses

8.2.8 Softwarelösungen für jede Komponente

Abbildung 85 zeigt die eingesetzten Technologien in SAFEWATER anhand der Systemsicht von RA-FWS^{NG}.

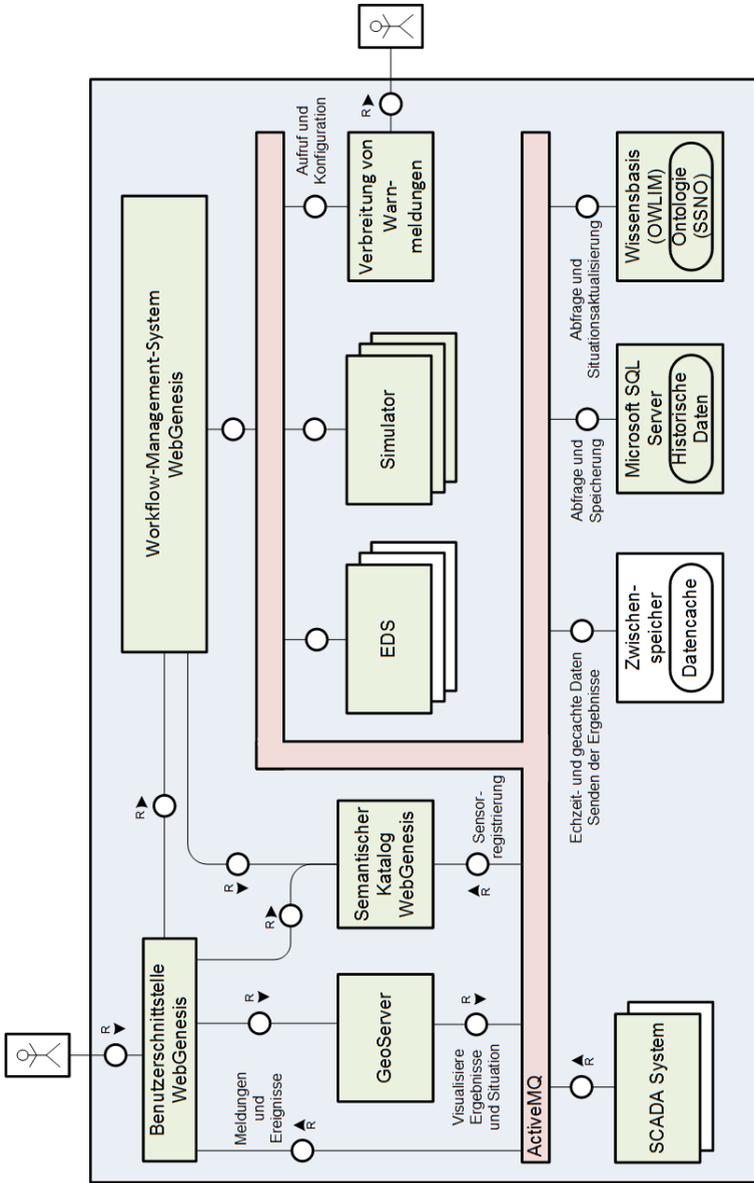


Abbildung 85: Eingesetzte Technologien für RA-FWS^{NG} in SAFEWATER

Zur Umsetzung der MOM wurde Apache ActiveMQ ausgewählt (s. Abschnitt 5.5.1.3). Als Datenquelle dient das jeweilige SCADA System des Wasserversorgers (s. Abschnitt 8.2.2). Die Daten werden in einem Microsoft SQL Server gehalten. Die Detektion der Frühwarnparameter wird durch den Verarbeitungsdienst EDS übernommen (s. Abschnitt 8.2.3), die Zwischenspeicherkomponente wurde nicht benötigt. Als Wissensbasis wurde OWLIM [Bishop, 2011] mit der SSNO eingesetzt.

Die Benutzeroberfläche und der semantische Katalog wurden mit WebGenesis® realisiert (s. Abschnitt 6.2.2). Da WebGenesis® bereits eine Workflow-Engine enthält, wurde diese als BPMS eingesetzt. Im Kern ist WebGenesis® außerdem ein Dokumentenmanagementsystem, sodass Dokumente der Domäne einfach bereitgestellt, recherchiert und bei Bedarf semantisch verknüpft werden können.

Die Software wurde bei drei großen Wasserversorgern (jeder versorgt ca. 1,5 Millionen Menschen) installiert und befindet sich momentan im Testbetrieb.

9 Bewertung und Schlussbetrachtungen

9.1 Bewertung RA-FWS^{NG}

Das in dieser Arbeit entwickelte Rahmenwerk RA-FWS^{NG} baut auf den Erfahrungen existierender Architekturen und Projekte auf, die in Kapitel 1 vorgestellt wurden (s. Abbildung 86), ergänzt sie jedoch um Lösungen für Anforderungen, die sich aus aktuellen Ergebnissen der Resilienzforschung ergeben und eine neue Qualität von Frühwarnsystemen fordern.

Im Kern stellt RA-FWS^{NG} eine Event-Driven SOA dar. Dieser Ansatz wurde gewählt, da er die Kombination der Vorteile von EDA und SOA ermöglicht (s. Abschnitt 3.1.4).

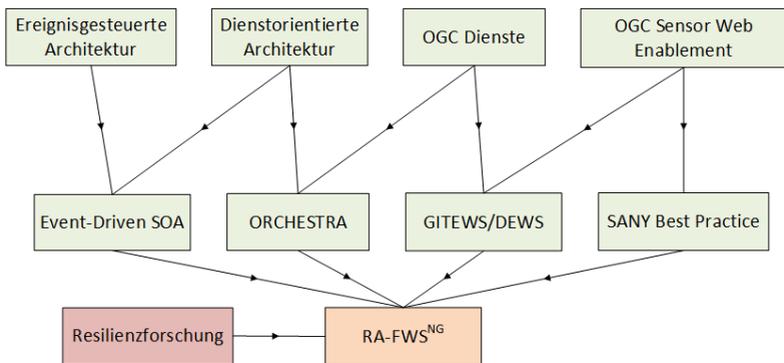


Abbildung 86: Architekturen und Projekte, die RA-FWS^{NG} beeinflussen

Die Orchestrierung von OGC Diensten wurde sowohl in ORCHESTRA, als auch in GITEWS/DEWS für Verarbeitungsdienste erprobt. RA-FWS^{NG} erweitert dies

auf alle Systemkomponenten mittels eines BPMS mit der Ankopplung an eine MOM. Von der in GITEWS/DEWS und SANY implementierte OGC SWE Architektur finden die Ideen der semantischen Sensordatenintegration Verwendung. GITEWS/DEWS verwendet auf Sensorebene eine MOM, RA-FWS^{NG} erweitert dies auf weitere Systemkomponenten.

Die folgenden beiden Tabellen enthalten die in Kapitel 1 aufgestellten Anforderungen. Die neu hinzugefügte Spalte *Lösung* verweist auf den entsprechenden Abschnitt, in dem die Vorgehensweise mit RA-FWS^{NG} zur Adressierung dieser Anforderung beschrieben wurde.

9.1.1 Funktionale Anforderungen

Tabelle 43 zeigt die Abdeckung der funktionalen Anforderungen durch RA-FWS^{NG}.

Tabelle 43: Abdeckung der funktionalen Anforderungen durch RA-FWS^{NG}

#	Anforderung	Lösung
	Generische Sensorintegration	
FA-1	- statisch	5.3.10, 5.5.9 und 6.2
FA-2	- dynamisch	5.3.10, 5.5.9 und 6.2
FA-3	- alternative Datenquellen (Soziale Medien, Crowdsourcing, Informationssysteme, etc.)	5.3.10, 5.5.9, 6.2 und 8.1.5
	Sensormanagement	
FA-4	- Katalog	5.3.10, 5.5.9 und 6.2
FA-5	- Semantischer Katalog	5.3.10, 5.5.9 und 6.2
FA-6	Integration heterogener Sensordaten	5.4, 6.1 und 6.4.2
	Generische Integration von Algorithmen	
FA-7	- statisch	5.3.6, 5.3.10, 5.5.9, 6.2 und 7.3
FA-8	- dynamisch	5.3.6, 5.3.10, 5.5.9, 6.2 und 7.3
FA-9	Dynamische Integration von Simulationen	5.3.10, 5.5.9, 6.2 und 8.2.5

FA-10	Integration von Standard Operating Procedures	5.3.9, 5.5.8 und 1
FA-11	Geographisches Informationssystem	5.3.11, 5.5.8 und 8.2.6
	Warnmeldungen	
FA-12	- personalisiert	5.3.8, 5.5.7 und 6.5
FA-13	- mehrsprachig	5.3.8, 5.5.7 und 6.5
FA-14	- mehrere und alternative Kommunikationswege	5.3.8, 5.5.7 und 6.5
FA-15	Recherche der Wissensbasis und von Dokumenten	6.6, 5.5.11 und 8.2.8
FA-16	Flexible Integration von Akteuren	5.1.3, 7.1
FA-17	Integrierte Benutzeroberfläche	5.3.12, 8.2.7
FA-18	Aggregation von Ereignissen	8.2.7
FA-19	Parallele Simulationen	8.2.5

9.1.2 Nichtfunktionale Anforderungen

Tabelle 44 zeigt die Abdeckung der nichtfunktionalen Anforderungen durch RA-FWS^{NG}.

Tabelle 44: Abdeckung nichtfunktionaler Anforderungen bei RA-FWS^{NG}

#	Anforderung	Lösung
NFA-1	Erstellung einer skalierbaren Kommunikationsinfrastruktur für ein SoS	5.2.2.1 und 8.1.3
NFA-2	Erstellung einer resilienten Kommunikationsinfrastruktur für SoS	5.2.2.2 und 8.1.3
NFA-3	Effiziente Verwaltung großer Datenmengen	5.3.6.4, 5.5.4 und 6.1.2
NFA-4	Skalierbare Speicherung großer verteilter Datenmengen	5.3.6.4, 5.5.4 und 6.1.2
NFA-5	Sicherheit in verteilten Systemen	5.6.3

9.1.3 Abgleich mit dem Stand von Forschung und Technik

In Tabelle 45 sind die von RA-FWS^{NG} abgedeckten Anforderungen an die Architekturen der Frühwarnsysteme, die dem Stand von Forschung und Technik entsprechen, gespiegelt. Dargestellt werden OGC SWE (s. Abschnitt 3.2.2), PRESTo (s. Abschnitt 3.4.2.1), GITEWS/DEWS (s. Abschnitt 3.4.3.1), TEWS (s. Abschnitt 3.4.3.2) und Flood EWS (s. Abschnitt 3.4.4.1).

Tabelle 45: Anforderungsabdeckung ausgewählter Frühwarnsysteme

#	Anforderung	SWE	PRESTo	DEWS	TEWS	EWS
	Generische Sensorintegration					
FA-1	- statisch	x	x	x	(x)	x
FA-2	- dynamisch	x		x	(x)	x
FA-3	- alternative Datenquellen	x		x		
	Sensormanagement					
FA-4	- Katalog	x	x	x	(x)	
FA-5	- semantischer Katalog					
FA-6	Integration heterogener Sensordaten	x	x	x	(x)	x
	Generische Integration von Algorithmen					
FA-7	- statisch					

#	Anforderung	SWE	PRESTo	DEWS	TEWS	EWS
FA-8	- dynamisch					
FA-9	Dynamische Integration von Simulationen					
FA-10	Integration von Standard Operating Procedures					
FA-11	Geographisches Informationssystem			x		x
	Warnmeldungen					
FA-12	- personalisiert		x	x		
FA-13	- mehrsprachig					
FA-14	- mehrere und alternative Kommunikationswege			x		
FA-15	Recherche der Wissensbasis und von Dokumenten					
FA-16	Flexible Integration von Akteuren					
FA-17	Integrierte Benutzeroberfläche		x	x		
FA-18	Aggregation von Ereignissen					
FA-19	Parallele Simulationen					
NFA-1	Erstellung einer skalierbaren Kommunikationsinfrastruktur für ein SoS		(x)	(x)		x
NFA-2	Erstellung einer resilienten Kommunikationsinfrastruktur für SoS		(x)	(x)		x
NFA-3	Effiziente Verwaltung großer Datenmengen					
NFA-4	Skalierbare Speicherung großer verteilter Datenmengen					
NFA-5	Sicherheit in verteilten Systemen		(x)	(x)		(x)

Die geklammerten Optionen bei NFA-1 und NFA-2 bedeuten, dass eine Infrastruktur auf Sensorebene vorhanden ist. Bei NFA-5 ist bei Systemen mit

geklammerter Option ein zentraler abgesicherter Zugang zum System vorhanden, da die Komponenten nicht verteilt sind. Inwiefern das verteilte Sensornetzwerk abgesichert ist, lässt sich aus den verfügbaren Veröffentlichungen nicht erkennen. Die Anforderungen FA-1 bis FA-6 wurden bei TEWS eingeklammert, da das hierfür eingesetzte SCADA System im Wesentlichen nur Prozessvariablen und die zugehörigen Daten kennt.

9.2 Zusammenfassung

Ziel der vorliegenden Arbeit war es, ein Rahmenwerk für die Architektur zukünftiger Frühwarnsysteme zu entwerfen. Dazu mussten zunächst die wesentlichen Architekturprobleme von Frühwarnsystemen identifiziert werden. Aus diesem Verständnis heraus wurden die Anforderungen an eine Rahmenarchitektur für Frühwarnsysteme definiert.

Danach wurde ein Rahmenwerk für die Systemarchitektur von Frühwarnsystemen mit Namen RA-FWS^{NG} entwickelt. Als Basis diente *RM-ODP*, ein Referenzmodell zur Beschreibung von verteilten Informationssystemen. Das Modell arbeitet mit verschiedenen Sichten auf ein System. Diese Sichten wurden für RA-FWS^{NG} ausgearbeitet und in Kapitel 1 ausführlich vorgestellt. Da es sich bei RA-FWS^{NG} um ein Rahmenwerk handelt, bleiben diverse Freiheitsgrade offen, die für eine konkrete Architektur eines Frühwarnsystems noch ausgearbeitet werden müssen. Dazu enthält jede Sicht eine Aufgabenliste der durchzuführenden Schritte mit entsprechenden Handlungsempfehlungen.

Semantische Technologien wurden zur Lösung verschiedenster Architekturprobleme vorgeschlagen, zum Beispiel die semantische Modellierung der Domäne, die Verwaltung der Sensorik, die Zusammenführung heterogener Daten, die Speicherung des Wissens, die Navigation durch das Netz an Informationen und die Präsentation von Informationen. Zu all diesen genannten Beispielen wurden in Kapitel 1 Lösungen erarbeitet und beschrieben.

Sowohl in der Phase der Bewertung und Erkennung von Gefahren als auch in der Reaktionsphase eines Frühwarnsystems spielt die Automation von Arbeitsabläufen eine wichtige Rolle. Hier wird als Beitrag die Nutzung von Methoden zur Geschäftsprozessmodellierung aus dem Unternehmensumfeld auf Frühwarnsysteme übertragen: Arbeitsabläufe werden mittels einer Sprache zur Geschäftsprozessmodellierung, der *Business Process Model and Notation (BPMN)*, modelliert und durch ein Workflow-Management-System interpretiert. Die Benutzeroberfläche stellt den Zugang zu einem integrierten Ende-zu-Ende System zur Verfügung, das den Umgang mit parallelen disruptiven Ereignissen ermöglicht.

Die Nutzbarkeit des Rahmenwerks RA-FWS^{NG} in der Praxis wurde abschließend in zwei Domänen erprobt und nachgewiesen: in der Frühwarnung vor Tsunamis und in der Frühwarnung vor verunreinigtem Trinkwasser. Besonders hervorzuheben ist die erleichterte Zusammenarbeit mit Fachexperten beim konkreten Systementwurf durch den Einsatz von Ontologien zur Modellierung des Domänenwissens und BPMN für SOPs.

9.3 Ausblick

Bei der Architektur eines Frühwarnsystems wird man mit der gesamten Bandbreite von Problemen eines hochkomplexen verteilten Systems konfrontiert. Aus der aktuellen Forschung in den Bereichen der Analyse großer Datenmengen, dem Internet der Dinge und der Informationsgewinnung aus sozialen Netzwerken ergeben sich jede Menge neue Ideen und Technologien. Diese müssen wiederum an den Anforderungen von Frühwarnsystemen reflektiert werden und können ggf. zu deren Verbesserung beitragen.

Eine wichtige Aufgabe der Forschung wird es sein, kostengünstige Möglichkeiten aufzuzeigen, um die notwendigen Daten zu sammeln und zusätzliche Informationsquellen zu integrieren. Insbesondere ein geringerer Energieverbrauch und die Nahfeld-Kommunikation in Sensornetz-

technologien sind vielversprechende Ansätze für eine kosteneffektive und umfassende Datensammlung zur Verbesserung regionaler Prognosen von Naturgefahren. Dadurch können Frühwarnsysteme in Bereichen und Ländern eingesetzt werden, in denen es bisher aus Kostengründen nicht möglich war. Bürger bauen mit solch günstiger Sensorik bereits eigene Sensornetzwerke auf. Durch Integration dieser Datenquellen kann eine deutliche bessere Abdeckung von Datenerhebungen erreicht werden. Jedoch ergeben sich auch völlig neue Fragestellungen nach der Genauigkeit und Vertrauenswürdigkeit solcher Messwerte. Für diese Problematik müssen noch Lösungen erforscht werden.

Neben den technischen Aspekten muss die zukünftige Forschung auf dem Gebiet der Frühwarnsysteme verstärkt mit interdisziplinären und kooperativen Ansätzen durchgeführt werden, um politische, soziale und auch ethische Probleme zu berücksichtigen [Wenzel, 2014]. Aber auch die Diskussion der Fachanwender mit IT Experten muss intensiviert werden, um die Nutzungsmöglichkeiten neuer Technologien für Frühwarnsysteme zu diskutieren und in Handlungsempfehlungen umzusetzen. Durch seine flexiblen Integrationsmöglichkeiten von Sensorik und Verarbeitungslogik und einer Verknüpfung zum Wissensmanagement ist das vorgestellte Architekturrahmenwerk bereits auf diese neuen Aufgaben ausgelegt und kann als Basis für Frühwarnsysteme der nächsten Generation dienen.

Abkürzungsverzeichnis

AMQP	Advanced Message Queuing Protocol
ANSI	American National Standards Institute
API	Application programming interface (Programmierschnittstelle)
BPEL	WS-Business Process Execution Language
BPM	Business Process Management (Prozessmanagement)
BPMN	Business Process Model and Notation
BPMS	Business Process Management System
CAP	OASIS Common Alerting Protocol; nicht zu verwechseln mit dem CAP-Theorem
CAP-Theorem	Consistency, Availability, Partition Tolerance - Theorem
CAT	OGC Catalogue Services
CEP	Complex Event Processing
CSW	OGC Catalogue Service for the Web
DMN	Decision Model and Notation
DSS	Decision Support System (Entscheidungsunterstützungssystem)
DUL	DOLCE + DnS Ultralite Ontology
EDA	Event-driven Architecture (ereignisgesteuerte Architektur)
EWS	Early Warning System (Frühwarnsystem)
RA-FWS ^{NG}	Early Warning System Architecture
FMC	Fundamental Modeling Concepts
FOAF	Friend of a Friend Ontology

GIS	Geographical information system (Geographisches Informationssystem)
GITEWS	German Indonesian Tsunami Early Warning System
GML	OGC Geography Markup Language: XML-format for geographical information
http	Hypertext Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
IIC	Industrial Internet Consortium
IoT	Internet of Things (Internet der Dinge)
IoT-A	Internet of Things – Architecture
IP	Internet Protocol
IRI	Internationalized Resource Identifier
ISO	International Organization for Standardization (Internationale Organisation für Normung)
JMS	Java Message Service
LAN	Local Area Network (lokales Netzwerk)
M2M	Machine-to-Machine Communication
MOM	Message-oriented middleware
MQTT	Message Queue Telemetry Transport
NetCDF	Network Common Data Form
O&M	OGC Observations and Measurements
OASIS	Organization for the Advancement of Structured Information Standards
OGC	Open Geospatial Consortium
OMG	Object Management Group
OPC-UA	OPC Unified Architecture

OPeNDAP	Open-source Project for a Network Data Access Protocol
OWL	Web Ontology Language
QoS	Quality of Service (Dienstgüte)
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
REST	Representational State Transfer
RIF	Rule Interchange Format
RM-OA	Reference Model for the ORCHESTRA Architecture
RM-ODP	Reference Model of Open Distributed Processing
SCADA	Supervisory Control and Data Acquisition
SensorML	OGC Sensor Model Language
SES	OGC Sensor Event Service
SOA	Service-Oriented Architecture (Serviceorientierte Architektur)
SOP	Standard Operating Procedure
SoS	System-of-Systems (System von Systemen)
SOS	OGC Sensor Observation Service
SPARQL	Simple Protocol and RDF Query Language
SPS	OGC Sensor Planning Service
SQL	Structured Query Language
SSNO	Semantic Sensor Network Ontology
SWE	OGC Sensor Web Enablement
SWEET	Semantic Web for Earth and Environmental Terminology
SWRL	Semantic Web Rule Language

TOGAF	Open Group Architecture Framework
UCUM	Unified Code for Units of Measure
UDDI	Universal Description Discovery and Integration
UML	Unified Modeling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
VCA	Vulnerability and capacity assessment (Bewertung der Verwundbarkeit und Leistungsfähigkeit)
W3C	World Wide Web Consortiums
WCMS	Web Content Management System
WFS	Web Feature Service
WMS	Web Map Service
WNS	OGC Web Notification Service
WWW	World Wide Web
XML	Extensible Markup Language

Abbildungen

Abbildung 1: Forschungsbereiche, die in die Arbeit einfließen (unterschiedliche Farben), mit Kapitelübersicht (Zahlen in Kreisen).....	10
Abbildung 2: Verwundbarkeit, Gefahr und Katastrophe nach [Blaikie, 2003].....	15
Abbildung 3: Phasen des Katastrophenmanagements [Austrian, 2011].....	18
Abbildung 4: Der Kreislauf des Katastrophenmanagements [NEHRP, 2016].....	19
Abbildung 5: Parallelität der Phasen des Katastrophenmanagements [FEMA, 2009].....	20
Abbildung 6: Krisenkreislauf nach [ZKI, 2013].....	22
Abbildung 7: Der Social Resilience Cycle [Edwards, 2009].....	24
Abbildung 8: Resilienz-Zyklus nach [Trachsler, 2009].....	25
Abbildung 9: Das Vier-Phasen Modell der Frühwarnung nach [Basher, 2006].....	27
Abbildung 10: Klassifikation von Systemeigenschaften nach [Richling, 2011].....	36
Abbildung 11: Klassifikation Nichtfunktionaler Systemeigenschaften nach [Avižienis, 2004], [Pohl, 2008], [ISO, 2011].....	37
Abbildung 12: Oberkategorien von Frühwarnsystemen.....	42
Abbildung 13: SoS-Architektur- und Entwicklungszyklen nach [Garlock, 2001].....	48
Abbildung 14: System, Architektur, Architekturmuster und Eigenschaft nach [Richling, 2011].....	61
Abbildung 15: Qualitätsbaum mit Szenarien [Starke, 2008].....	63

Abbildung 16: Themenbereich der Architekturbeschreibung	64
Abbildung 17: Beispiel für FMC Blockdiagramm.....	67
Abbildung 18: Der Semantic Web Stack nach [Berners-Lee, 2000]	73
Abbildung 19: Ein einfacher Prozess.....	82
Abbildung 20: Symbole für verschiedene Arten von Aktivitäten.....	83
Abbildung 21: Das Symbol für einen Sequenzverlauf	83
Abbildung 22: Symbole für die verschiedenen Ereignistypen	84
Abbildung 23: Symbole für verschiedene Arten von Gateways.....	84
Abbildung 24: Ein Pool mit zwei Lanes	85
Abbildung 25: Nachrichtenaustausch zwischen Pools.....	86
Abbildung 26: Sichten in RM-ODP nach [Dekker, 2008]	90
Abbildung 27: Dienste in dienstorientierten Architekturen [Gebhart, 2011]	94
Abbildung 28: Kombination der Konzepte von EDA und SOA nach [Buckel, 2012]	99
Abbildung 29: Lambda Architektur nach Marz [Marz, 2015].....	100
Abbildung 30: Systemsicht der IoT Reference Architecture (IoT-A) [Carrez, 2013]	103
Abbildung 31: Funktionale Sicht der IIC Architektur [IIC, 2015]	105
Abbildung 32: Interaktion der SWE Dienste und Kodierungen - Teil 1 [Simonis, 2008]	106
Abbildung 33: Interaktion der SWE Dienste und Kodierungen - Teil 2 [Simonis, 2008]	107
Abbildung 34: Einsatzbeispiel für SWE Dienste [Bröring, 2011]	108
Abbildung 35: Die SANY Schichtenarchitektur [Usländer, 2009]	109
Abbildung 36: Beispiel für SANY Verarbeitungskette mittels OGC- Diensten [Middleton, 2010]	110

Abbildung 37: Komponenten eines Entscheidungsunterstützungs- systems [Pinto, 2014]	113
Abbildung 38: Komponenten eines Entscheidungsunterstützungs- systems [Bröring, 2011]	114
Abbildung 39: Architektur für Expertensysteme nach [Bröring, 2011].....	115
Abbildung 40: EWS als Schichtenarchitektur [Meissen, 2013]	116
Abbildung 41: Subsysteme eines Ende-zu-Ende EWS [Meissen, 2006]	118
Abbildung 42: Architektur zur generischen Integration von Human Sensors [Meissen, 2014a]	120
Abbildung 43: Architektur zur Verbreitung von Warnmeldungen nach [Montanari, 2007]	122
Abbildung 44: PRESto Architektur und Datenfluss [Satriano, 2011]	125
Abbildung 45: GITEWS / DEWS Grobarchitektur [Pinto, 2014].....	127
Abbildung 46: GITEWS Architektur mit Tsunami Sensor Bus (TSB) [Fleischer, 2010]	128
Abbildung 47: Flutwarnsystem basierend auf CIS [Balis, 2011].....	131
Abbildung 48: Zuständigkeiten im Katastrophenschutz in Baden- Württemberg nach [Bell, 2010]	148
Abbildung 49: Informelle Darstellung einer SOP nach [Bell, 2010]	155
Abbildung 50: SoS mit drei angeschlossenen Standorten	157
Abbildung 51: Systemsicht RA-FWS ^{NG}	160
Abbildung 52: Ablauf und Interaktion von Verarbeitungsdiensten	164
Abbildung 53: Aufbau und Interaktion der Komponente zur Verbreitung von Warnmeldungen.....	169
Abbildung 54: Integration der GIS Komponente in RA-FWS ^{NG}	172
Abbildung 55: Bestandteile einer O&M Beobachtung [Bröring, 2011]	176
Abbildung 56: Datenelemente von CAP.....	185
Abbildung 57: Einsatz semantischer Technologien in RA-FWS ^{NG}	196

Abbildung 58: Transformations- und Analyseprozess zur Datenintegration	198
Abbildung 59: Abbildung mehrerer Schnittstellen auf ein existierendes Informationssystem	200
Abbildung 60: Aufbau des semantischen Katalogs und Integration in RA-FWS ^{NG}	203
Abbildung 61: In WebGenesis® importierte Konzepte aus SSNO und DUL ..	205
Abbildung 62: Hierarchie, Beschreibungstext und Eigenschaften des SSNO Konzepts Sensor	206
Abbildung 63: Automatisch generiertes Eingabeformular für das SSNO Konzept Sensor	207
Abbildung 64: Editor für semantische Suchen	208
Abbildung 65: Aus der semantischen Suche generiertes Suchformular	209
Abbildung 66: Hauptkonzepte SSNO und ihre Beziehungen	214
Abbildung 67: Wie greifen Dienste in einer verteilten Umgebung auf die Wissensbasis zu?	217
Abbildung 68: Meldung für Person, mit Interesse an Feinstaubbelastung in der Innenstadt.....	222
Abbildung 69: Meldung für Person, die eine Wanderung unternehmen möchte	222
Abbildung 70: Die gleiche Meldung in Englisch, Finnisch und Schwedisch ..	223
Abbildung 71: Ablauf der mehrsprachigen Meldungsgenerierung.....	224
Abbildung 72: Aufbau des Workflow-Management-Systems mit integriertem Regelinterpreter	229
Abbildung 73: Ablauf der Integration des Regelinterpreters	230
Abbildung 74: Verschiedene Hilfstasks für BPMN	231
Abbildung 75: Ablauf der semi-automatisierten Tsunami-Frühwarnung ..	235
Abbildung 76: Abgleich des Szenarios mit Fehlerbehandlung.....	236
Abbildung 77: Eingesetzte Technologien für RA-FWS ^{NG} in TRIDEC.....	240

Abbildung 78: Einfacher Ablauf des Ereignismanagements	243
Abbildung 79: BPMN Prozess zur Deklaration eines Ereignisses	244
Abbildung 80: BPMN Prozess zur Reaktionsphase	245
Abbildung 81: Integration von Simulatoren.....	250
Abbildung 82: Simulationsergebnis einer Ausbreitungsberechnung.....	251
Abbildung 83: Ansicht der Alarme	253
Abbildung 84: Deklaration eines Ereignisses	254
Abbildung 85: Eingesetzte Technologien für RA-FWS ^{NG} in SAFEWATER....	255
Abbildung 86: Architekturen und Projekte, die RA-FWS ^{NG} beeinflussten .	257

Tabellen

Tabelle 1: Aufbau einer SOP	23
Tabelle 2: Beschreibung der Systemeigenschaften	38
Tabelle 3: Techniken von GIS	51
Tabelle 4: OGC SWE Standards.....	53
Tabelle 5: Weitere OGC Standards.....	55
Tabelle 6: Vorlaufzeiten für die Frühwarnung [IFRC, 2012]	58
Tabelle 7: Entscheidungstabelle mit beschränkten Einträgen	87
Tabelle 8: Entscheidungstabelle mit erweiterten Einträgen	87
Tabelle 9: Bewertung SOA.....	95
Tabelle 10: Bewertung Microservices	96
Tabelle 11: Bewertung EDA.....	98
Tabelle 12: Bewertung Lambda Architektur.....	101
Tabelle 13: Bewertung von IoT-A	104
Tabelle 14: Bewertung SWE	108
Tabelle 15: Bewertung SANY Architektur	112
Tabelle 16: Bewertung RA-EWS.....	119
Tabelle 17: Bewertung PWS	123
Tabelle 18: Bewertung PRESTO.....	124
Tabelle 19: Bewertung GITEWS/DEWS.....	129
Tabelle 20: Bewertung TEWS	130
Tabelle 21: Bewertung Flood EWS	133
Tabelle 22: Datenquellen für Katastrophen [Ionita, 2014]	136
Tabelle 23: Funktionale Anforderungen an RA-FWS ^{NG}	145
Tabelle 24: Nichtfunktionale Anforderungen an RA-FWS ^{NG}	146

Tabelle 25: Arbeitsschritte der Unternehmenssicht	154
Tabelle 26: Prüfung auf SoS am Beispiel Tsunami-Frühwarnung.....	156
Tabelle 27: Arbeitsschritte der Sicht System von Systemen	159
Tabelle 28: Arbeitsschritte der Systemsicht	174
Tabelle 29: Arbeitsschritte der Informationssicht.....	178
Tabelle 30: Standards zum Zugriff auf Sensordaten	180
Tabelle 31: Arbeitsschritte der Technologiesicht	187
Tabelle 32: Integrationsmatrix RA-FWS ^{NG} mit Testfällen	188
Tabelle 33: Integrationstests Phase 1	189
Tabelle 34: Integrationstests Phase 2	190
Tabelle 35: Integrationstests Phase 3	190
Tabelle 36: Arbeitsschritte der Konstruktionssicht	192
Tabelle 37: Durchzuführende Arbeitsschritte für RA-FWS ^{NG}	192
Tabelle 38: Befehlssatz der MOM- und REST-Schnittstelle	204
Tabelle 39: Relevante Ontologien	211
Tabelle 40: Schnittstelle zum Knowledge Base Access Service (KBAS)	219
Tabelle 41: Vorgestellte Arbeitsschritte für den Anwendungsfall Tsunami ..	234
Tabelle 42: Vorgestellte Schritte für Sicheres Trinkwasser	242
Tabelle 43: Abdeckung der funktionalen Anforderungen durch RA-FWS ^{NG} ..	258
Tabelle 44: Abdeckung nichtfunktionaler Anforderungen bei RA-FWS ^{NG} ...	260
Tabelle 45: Anforderungsabdeckung ausgewählter Frühwarnsysteme	260

Literaturverzeichnis (alphabetisch)

- [Addams-Moring, 2005] R. Addams-Moring, M. Kekkonen, S. Zhao „A Simple Taxonomy for Mobile Emergency Announcement Systems”. Proceedings of the 2nd International ISCRAM Conference, S. 309–316, 2005
- [Allen, 2004] J. F. Allen „Actions and Events in Interval Temporal Logic”. Journal of Logic and Computation, vol. 4, S. 531-579., 2004
- [Allweyer, 2014] T. Allweyer „BPMS: Einführung in Business Process Management-Systeme”. 1. Auflage, Books on Demand, 2014
- [Ammann, 2016] W. J. Ammann (Ed.) „Integrative Risk Management - Towards Resilient Cities”. Proceedings of IDRC DAVOS 2016, 2016
- [ANSS, 2002] ANSS Technical Integration Committee „Technical Guidelines for the Implementation of The Advanced National Seismic System”. U.S. Department of the Interior, U.S. Geological Survey, 2002
- [Aronson, 2001] J. Aronson, E. Turban „Decision Support Systems and Intelligent Systems”. Upper Saddle River, NJ: Prentice-Hall, 2001
- [Austrian, 2011] Austrian Standards „Integriertes Katastrophenmanagement - Benennungen und Definitionen”. ÖNORM S 2304, 2011
- [Avižienis, 2001] A. Avižienis, J.-C. Laprie, B. Randell „Fundamental Concepts of dependability”. Research Report N01145, LAAS-CNRS, 2001
- [Avižienis, 2004] A. Avižienis, J.-C. Laprie, B. Randell, C. Landwehr „Basic Concepts and Taxonomy of Dependable and Secure Computing”. IEEE Transactions on dependable and secure computing, Vol. 1, No. 1, 2004
- [Baird, 2010] M. E. Baird „The *Phases* of Emergency Management – Background Paper”. Intermodal Freight Transportation Institute (ITFI), University of Memphis, 2010
- [Baldwin, 2008] K. J. Baldwin „Systems Engineering Guide for Systems of Systems, Version 1.0”. Washington, DC: ODUSD (A&T) SSE, 2008

- [Balis, 2011] B. Balis, M. Kasztelnik, M. Bubak, T. Bartynski, T. Gubala, P. Nowakowski, J. Broekhuijsen „The UrbanFlood Common Information Space for Early Warning Systems”. *Procedia Computer Science* V.4, S. 96-105, DOI: 10.1016/j.procs.2011.04.011, 2011
- [Banks, 2014] A. Banks, R. Gupta (Eds.) „MQTT Version 3.1.1”. OASIS Standard, 29. Oktober, 2014
- [Bapat, 1994] S. Bapat „Object-Oriented Networks, Models for Architecture, Operations and Management”. Englewood Cliffs, 1994
- [Bartelme, 1989] N. Bartelme „GIS-Technologie. Geoinformationssysteme, Landinformationssysteme und ihre Grundlagen”. Springer, 1989
- [Bartelme, 2000] N. Bartelme „Geoinformatik - Modelle, Strukturen, Funktionen”. 3. Auflage, Springer, 2000
- [Basher, 2006] R. Basher, J. Bogardi, S. Dannenmann, J. C. Villagran de León “Early Warning Systems in the context of Disaster Risk Management”. *Entwicklung & Ländlicher Raum* 2/2006, 2006
- [Bauernhansl, 2014] T. Bauernhansl, M. Ten Hompel, B. Vogel-Heuser (Eds.) „Industrie 4.0 in Produktion, Automatisierung und Logistik: Anwendung - Technologien - Migration”. Springer-Verlag, 2014
- [Bell, 2010] R. Bell, J. Mayer, J. Pohl, S. Greiving, T. Glade „Integrative Frühwarnsysteme für gravitative Massenbewegungen (ILEWS). Monitoring, Modellierung, Implementierung”. Klartext Verlag, Essen, 2010
- [Bellahsene, 2011] Z. Bellahsene, A. Bonifati, E. Rahm „Schema Matching and Mapping”. Springer Verlag, DOI: 10.1007/978-3-642-16518-4, 2011
- [Bellwood, 2002] T. Bellwood „UDDI Version 2.04 API Specification”. OASIS, 2002
- [Bengel, 2002] G. Bengel „Verteilte Systeme: Client-Server-Computing für Studenten und Praktiker”. 2. Auflage, Braunschweig, 2002
- [Berners-Lee, 2000] T. Berners-Lee „Semantic Web on XML”. XML 2000 Washington DC, <https://www.w3.org/2000/Talks/1206-xml2k-tbl>, , letzter Abruf am 24.9.2016, 2000

- [Bernstein, 2011] P. Bernstein, J. Madhavan, E. Rahm "Generic Schema Matching, Ten Years Later". Proceedings of 37th International Conference on Very Large Data Bases, Seattle, Washington, 2011
- [Berntheisel, 2009] S. Berntheisel „Geschäftsprozessmodellierung mit BPEL“. Seminararbeit , Hochschule RheinMain, 2009
- [Bishop, 2011] B. Bishop, A. Kiryakov, D. Ognyanoff, I. Peikov, Z. Tashev, R. Velkov, „OWLIM: A family of scalable semantic repositories“. Semantic Web, Volume 2, Nummer 1, 2011
- [BITKOM, 2014] Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e. V. „Big-Data-Technologien – Wissen für Entscheider – Leitfaden“. 2014
- [Blaikie, 2003] P. Blaikie, T. Cannon, I. Davis, B. Wisner „At Risk: Natural Hazards, People's Vulnerability and Disasters“. 2nd Edition, Routledge, 2003
- [Booch, 1998] G. Booch, J. Rumbaugh, I. Jacobson „The Unified Modeling Language User Guide“. ISBN 0-201-57168-4, Addison Wesley, 1998
- [Booch, 1999] G. Booch, I. Jacobson, J. Rumbaugh „The Unified Software Development Process“. Addison-Wesley, 1999
- [Botterell, 2006] A. Botterell „The Common Alerting Protocol: An Open Standard for Alerting, Warning and Notification“. Proceedings of the 3rd International ISCRAM Conference, 2006
- [Botts, 2008] M. Botts, G. Percivall, C. Reed, J. Davidson „OGC Sensor Web Enablement: Overview and High Level Architecture“. Proceedings of 5th International ISCRAM Conference, 2008
- [Bouras, 2010] A. Bouras, P. Gouvas, A. Friesen, S. Pantelopoulos, S. Alexakis, G. Mentzas „Business process fusion based on semantically-enabled service-oriented business applications“. Proceedings of the Workshops and the Doctorial Symposium of the Second IFAC/IFIP I-ESA International Conference: EI2N, WSI, IS-TSPQ 2006, Vol. 119, S. 157, John Wiley & Sons, 2010
- [Bröhl, 1995] A. P. Bröhl, W. Dröschel „Das V-Modell: Der Standard für die Softwareentwicklung mit Praxisleitfaden“. Oldenbourg Wissenschaftsverlag, 1995

- [Broekstra, 2002] J. Broekstra, A. Kampman, F. van Harmelen „Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema”. Proceedings of International Semantic Web Conference, S. 54-68, 2002
- [Bröring, 2011] A. Bröring, J. Echterhoff, S. Jirka, I. Simonis, T. Everding, C. Stasch, S. Liang, R. Lemmens „New Generation Sensor Web Enablement”. Sensors 2011, 11, DOI: 10.3390/s110302652, 2011
- [Bruns, 2010] R. Bruns, J. Dunkel „Event-driven architecture: Softwarearchitektur für ereignisgesteuerte Geschäftsprozesse”. Springer-Verlag, 2010
- [Buckel, 2012] T. Buckel „Zum Potential von Event-Driven Architecture für komplexe Unternehmensnetzwerke”. Tagungsband der Multikonferenz Wirtschaftsinformatik 2012 (MKWI 2012), 2012
- [Bügel, 2011] U. Bügel, M. Schmieder, B. Schnebel, T. Schlachter, R. Ebel „Leveraging Ontologies for Environmental Information Systems”. Proceedings of ISESS 2011, IFIP AICT 359, S. 364-371, 2011
- [Burstein, 2008] F. Burstein, C. W. Holsapple „Handbook on Decision Support Systems 1 - Basic Themes”. Springer-Verlag, 2008
- [Caroll, 2004] J. J. Caroll, I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne, K. Wilkinson „Jena: implementing the semantic web recommendations”. Proceedings of the 13th international World Wide Web conference, S. 74-83, 2004
- [Carrez, 2013] F. Carrez (Ed.) „Final architectural reference model for the IoT v3.0”, Internet of Things – Architecture (IoT-A), Deliverable D1.5, 2013
- [Chang, 2006] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, R. E. Gruber „Bigtable: A distributed storage system for structured data”. Proceedings of the 7th conference on USENIX Symposium on Operating Systems Design and Implementation - Volume 7, 2006
- [Chen, 1976] P. P. Chen „The Entity-relationship Model—Toward a Unified View of Data”. ACM Trans. Database Syst. Band 1, Nr. 1, 1976
- [Chou, 2008] D. Chou „Using Events in Highly Distributed Architectures”. The Architecture Journal 17, S. 29-33, 2008

- [Christakos, 2001] G. Christakos „Modern spatiotemporal geostatistics”. Oxford University Press, New York, 2nd Edition, 2001
- [Clements, 2002] P. Clements, R. Kazman, M. Klein „Evaluating Software Architecture – Methods and Case Studies”. Addison-Wesley, 2002
- [Compton, 2012] M. Compton, P. Barnaghi, L. Bermudez, R. García-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog, V. Huang, K. Janowicz, W. D. Kelsey, D. Le Phuoc, L. Lefort, M. Leggeri, H. Neuhaus, A. Nikolov, K. Page, A. Passant, A. Sheth, K. Taylor „The SSN ontology of the W3C semantic sensor network incubator group”. Web Semantics: Science, Services, 2012
- [Cornillon, 2003] P. Cornillon, J. Gallagher, T. Sgouros „OPeNDAP: Accessing data in a distributed, heterogeneous environment”. Data Science Journal, Volume 2, 5. November, 2003
- [Coulouris, 2012] G. Coulouris, J. Dollimore, T. Kindberg, G. Blair „Distributed Systems - Concepts and Design”. Fifth Edition, Addison-Wesley, 2012
- [Dąbrowski, 2015] M. Dąbrowski „A GIS-based method for supporting operational events management in water supply system”. Proceedings of Sustainable Communities - Sharing Knowledge, 2015
- [Dean, 2004] J. Dean, S. Ghemawat „MapReduce: simplified data processing on large clusters”. Proceedings of the 6th conference on Symposium on Operating Systems Design & Implementation, USENIX Association, 2004
- [Dekker, 2008] M. Dekker. „Global illustration of the RM-ODP Viewpoints”. http://en.wikipedia.org/wiki/File:RM-ODP_viewpoints.jpg, letzter Abruf am 11.4.2016, 2016
- [Diekmann, 2003] T. Diekmann, S. Hagenhoff „Verteilte Systeme: State of the Art”. Arbeitsbericht Nr. 1/2003, Institut für Wirtschaftsinformatik, Göttingen, 2003
- [DIN, 1988] Deutsches Institut für Normung e.V. „Informationsverarbeitung - Begriffe - Teil 1: Allgemeine Begriffe”. DIN 44300-1, 1988
- [DoD, 1997] United States Department of Defense „Technical Architecture Framework for Information Management”. Vol. 1, April, 1996

- [DoD, 2009] United States Department of Defense „Department of Defense Architecture Framework (DoDAF) Version 2.0 — Volume 1: Introduction, Overview, and Concepts — Manager’s Guide“. 2009
- [Dokas, 2007] I. M. Dokas „Ontology to Support Knowledge Representation and Risk Analysis for the Development of Early Warning System in Solid Waste Management Operations“. In Environmental Software Systems - Dimensions of Environmental Informatics, 7th International Symposium on Environmental Software Systems (ISESS 2007) S. 109, 2007
- [Duden, 2011] Duden „Deutsches Universal Wörterbuch“. Bibliographisches Institut & FA Brockhaus AG, 2006
- [EABPM, 2011] European Association of Business Process Management (EABPM) „Business Process Management Common Body of Knowledge - BPM CBOK: Leitfaden für das Prozessmanagement“. Verlag Dr. Götz Schmidt, Wettenberg, 2011
- [Edwards, 2009] C. Edwards „Resilient Nation“. Demos, ISBN 978-1-906693-13-8, 2009
- [EFD, 2011] Eidgenössisches Finanzdepartement (EFD)
„Prozessdokumentation mittels BPMN am Beispiel von Meldeprozessen der AHV/IV. Umsetzungsbeispiel für den Standard eCH-0073
Dokumentation öffentlicher Leistungen und Prozesse“.
http://www.ech.ch/alfresco/guestDownload/attach/workspace/SpacesStore/f2acf3d0-6569-47b2-9405-9180a63c7375/BERI_d_DEF_2011-06-01_BPMN_eAHV.pdf, letzter Abruf am 17.6.2016, 2011
- [Eigner, 2012] M. Eigner, F. Gerhardt, T. Gilz, F. M. Nem
„Informationstechnologie für Ingenieure“. Springer-Verlag, 2012
- [Eicker, 2007] S. Eicker, C. Hegmanns, S. Malich „Auswahl von Bewertungsmethoden für Softwarearchitekturen“. ICB-Research Report 14, ISSN 1860-2770, März, 2007
- [ENVIROFI, 2012] ENVIROFI Consortium „Data and meta-information specifications - Technical Report“. DOI: 10.13140/RG.2.1.2094.4242, 2012

- [EO2HEAVEN, 2013] EO2HEAVEN Consortium „OGC Best Practices for Sensor Web Enablement: Provision of Observations through an OGC Sensor Observation Service (SOS)“. OGC document 13-015, 2013
- [Erl, 2008] T. Erl „SOA – Principles of Service Design“. Prentice Hall, ISBN 978-0-13-234482-1, 2008
- [Erling, 2009] O. Erling, I. Mikhailov, „Virtuoso: RDF Support in a Native RDBMS“. Semantic Web Information Management, Springer Berlin Heidelberg, S. 501-519, 2009
- [Ewert, 1997] G. Ewert, W. Umstätter „Lehrbuch der Bibliotheksverwaltung“. Hiersemann Verlag, 1997
- [FEMA, 2009] Federal Emergency Management Agency (FEMA) „Comprehensive Preparedness Guide, Developing and Maintaining State, Territorial, Tribal, and Local Government Emergency Plan“. CPG 101, Washington, 2009
- [Fertier, 2016] A. Fertier, A. Montarnal, A. Barthe-Delanoë, S. Truptil, F. Bénaben „Adoption of big data in crisis management toward a better support in decision-making“. Proceedings of the ISCRAM 2016 Conference, 2016
- [Fleischer, 2010] J. Fleischer, R. Häner, S. Herrnkind, A. Kloth, U. Kriegel, H. Schwarting, J. Wächter „An integration platform for heterogeneous sensor systems in GITEWS – Tsunami Service Bus“. Natural Hazards and Earth System Sciences, DOI: 10.5194/nhess-10-1239-2010, 2010
- [Forrester, 2013] Forrester „Webinar 2013 - Big Data: Gold Rush Or Illusion?“. http://solutions.forrester.com/Global/FileLib/webinars/Big_Data_-_Gold_Rush_or_Illusion.pdf, letzter Abruf am 22.5.2016, 2013
- [Forslund, 1995] G. Forslund „Toward Cooperative Advice-Giving Systems: A Case Study in Knowledge Based Decision Support“. IEEE Expert, 1995
- [Fowler, 2015] M. Fowler, J. Lewis "Microservices - Nur ein weiteres Konzept in der Softwarearchitektur oder mehr?". OBJEKTSpektrum 01/2015, 2015
- [French, 2013] S. French, N. Argyris, W. J. Nuttall, J. Moriarty, P. J. Thomas „The early phase of a radiation accident: Revisiting thinking on

- evacuation and exclusion zones". ISCRAM 2013 Conference Proceedings – 10th International Conference on Information Systems for Crisis Response and Management, 2013
- [Freund, 2010] J. Freund, B. Rücker „Praxishandbuch BPMN 2.0“. 2. Auflage, Hanser Verlag, 2010
- [Fritzsche, 2007] M. Fritzsche, P. Keil „Kategorisierung etablierter Vorgehensmodelle und ihre Verbreitung in der deutschen Software-Industrie“. Technische Universität München, Institut für Informatik, TUM-INFO-06-I0717-100/1.-FI, 2007
- [Gamma, 1996] E. Gamma, R. Helm, R. Johnson, J. Vlissides „Entwurfsmuster – Elemente wiederverwendbarer objektorientierter Software“. 1. Auflage, Addison-Wesley, 1996
- [Garlock, 2001] P. F. Garlock „System of Systems (SoS) - Enterprise Systems Engineering for Information-Intensive Organizations“. Systems Engineering, Vol. 4, No. 4, 2001
- [GDV, 2011] Gesamtverband der Deutschen Versicherungswirtschaft e.V. „Jahrbuch der deutschen Versicherungswirtschaft 2011“. GDV, 2011
- [Gebhart, 2011] M. Gebhart „Qualitätsorientierter Entwurf von Anwendungsdiensten“. Dissertation, Fakultät für Informatik am Karlsruher Institut für Technologie (KIT), 2011
- [Gerhold, 2011] L. Gerhold „Dokumentation des Workshops Naturereignisse & Naturkatastrophen“, Forschungsforum Öffentliche Sicherheit, 2011
- [Gilbert, 2002] S. Gilbert, N. Lynch „Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services“. ACM SIGACT News, V. 33 Issue 2, S. 51–59, 2002
- [Godfrey, 2012] R. Godfrey, D. Ingham, R. Schlomin (Eds.) „Advanced Message Queuing Protocol (AMQP) Version 1.0“. OASIS Standard, 29. Oktober, 2012
- [Gray, 2010] A. Gray (Ed.) „SemSorGrid4Env Architecture - Phase II“. EU FP7 Project SemSorGrid4Env - Semantic Sensor Grids for Rapid Application Development for Environmental Management,

- <http://www.sensorgid4env.eu/files/deliverables/wp1/D1.3v2.pdf>, letzter Abruf am 4.9.2016, 2010
- [Groot, 2006] W. de Groot, J. Goldammer, T. Keenan, M. Brady, T. Lynham, C. Justice, I. Csiszar, K. O'Loughlin „Developing a global early warning system for wildland fire”. *Forest Ecology and Management*, Vol. 234, no. 1, 2006
- [Gross, 2012] A. Gross, M. Hartung, A. Thor, A. Rahm „How do computed ontology mappings evolve?”. *Joint Workshop on Knowledge Evolution and Ontology Dynamics (ISWC)*, 2012
- [Gruber, 1995] T. Gruber „Toward principles for the design of ontologies used for knowledge sharing”. *International Journal of Human-Computer Studies*, Band 43(5-6), 1995
- [Gruninger, 2002] M. Gruninger, J. Lee „Ontology applications and design”. *Communications of the ACM*, 2002
- [Handschuh, 2005] S. Handschuh „Creating ontology-based metadata by annotation for the semantic web”. *Dissertation, Fakultät für Wirtschaftswissenschaften - Institut für Angewandte Informatik und Formale Beschreibungsverfahren (AIFB) Universität Karlsruhe*, 2005
- [Hanmer, 2007] R. S. Hanmer „Patterns for Fault Tolerant Software”. *Wiley*, 2007
- [Hapner, 1999] M. Hapner, R. Burridge, R. Sharma „Java™ Message Service Version 1.0.2”. *Sun Microsystems*, 9. November, 1999
- [Häring, 2016] I. Häring, B. Scharte, S. Hiermaier „Towards a novel and applicable approach for Resilience Engineering”. *Proceedings of IDRC 2016*, S. 272, 2016
- [He, 2007] X. He, Z. Ma, W. Shao, G. Li „A metamodel for the notation of graphical modeling languages”. *Proceedings of Computer Software and Applications Conference (COMPSAC)*, 2007
- [Heinrich, 2011] L. J. Heinrich, A. Heinzl, R. Riedl „*Wirtschaftsinformatik*”. 4. Auflage, *Springer-Verlag*, 2011

- [Hesse, 2008] W. Hesse, H. C. Mayr „Modellierung in der Softwaretechnik: eine Bestandsaufnahme“. Informatik-Spektrum, Vol. 31, No. 5., Oktober 2008
- [Hilbring, 2014] D. Hilbring, T. Titzschkau, A. Buchmann, G. Bonn, F. Wenzel, E. Hohnecker „Earthquake Early Warning for Transport Lines“. Natural Hazards, Volume 70, Issue 3, S. 1795-1825, 2014
- [Hindel, 2004] B. Hindel, K. Hörmann, M. Müller, J. Schmied „Basiswissen Software-Projektmanagement“. dpunkt.verlag, 2004
- [Hitzler, 2008] P. Hitzler, M. Krötzsch, S. Rudolph, Y. Sure „Semantic web: Grundlagen“ Springer-Online, 2008
- [Horn, 2016] T. Horn „SOA (Service Oriented Architecture) - Entscheidung für SOA“. <http://www.torsten-horn.de/techdocs/soa.htm>, letzter Abruf am 5.7.2016, 2016
- [Horridge, 2008] M. Horridge, S. Bechhofer „The OWL API: A Java API for working with OWL 2 ontologies“. OWLED, Volume 529, Proceedings of CEUR Workshop, 2008.
- [Hudelot, 2008] C. A. Hudelot „Fuzzy Spatial Relation Ontology for Image Interpretation, Fuzzy Sets and Systems“. 2008
- [IEEE, 1990] IEEE „Standard Glossary of Software Engineering Terminology“. Institute of Electrical and Electronics Engineers, 1990
- [IEEE, 2000] ANSI/IEEE „Recommended Practice for Architecture Description of Software-Intensive Systems“. ANSI/IEEE 1471-2000, 2000
- [IFRC, 1999] International Federation of Red Cross and Red Crescent Societies „Vulnerability and Capacity Assessment Guide“. <http://www.ifrc.org/>, letzter Abruf am 4.9.2016, 1999
- [IFRC, 2012] International Federation of Red Cross and Red Crescent Societies „Community early warning systems: guiding principles“. Geneva, 2012
- [IIC, 2015] Industrial Internet Consortium (IIC) „Industrial Internet Reference Architecture“. Version 1.7, 2015
- [Iofciu, 2005] T. Iofciu, C. Kohlschütter, W. Nejdli, R. Paiu „Keywords and rdf fragments: Integrating metadata and full-text search in beagle++“.

- Proceedings of Semantic Desktop Workshop at the ISWC, Galway, Ireland, 2005
- [Ionita, 2008] V. Ionita, A. D. Ionita, „Architecture for Integrating Data Obtained by Advanced Characterization of Magnetic Materials”, Romanian Journal of Materials, vol. 38 (1), S. 69-75, 2008.
- [Ionita, 2014] A. D. Ionita, A. Olteanu „Data Acquisition, Processing and Visualization for Early Warning Information Systems”. IEEE International Symposium on Fundamentals of Electrical Engineering, 2014
- [ISDR, 2016] International Strategy for Disaster Reduction (ISDR) „What's early warning”. Platform for the Promotion of Early Warning, <http://www.unisdr.org/2006/ppew/whats-ew/basics-ew.htm>, letzter Abruf am 5.5.2016, 2016
- [ISO, 1995] ISO/IEC JTC1/SC21 „Open Distributed Processing - Reference Model – Part 2: Foundations”. IS 10746-2/ITU-T Recommendation X.902, 1995
- [ISO, 2009] Joint Technical Committee ISO/IEC JTC 1 „Information technology - Open distributed processing - Reference model: Architecture”. ISO/IEC 10746-3, 2009
- [ISO, 2011] ISO „System und Software-Engineering – Qualitätskriterien und Bewertung von System und Softwareprodukten (SQuaRE) – Qualitätsmodell und Leitlinien”. ISO/IEC 25010:2011, 2011
- [ISO, 2015] ISO „Anforderungen an ein Qualitätsmanagementsystem (QM-System) ”. DIN EN ISO 9001, 2015
- [ISO, 2016] ISO „Geographic Information – Services”. ISO 19119:2016, 2016
- [Jeckle, 2004] M. Jeckler, C. Rupp, J. Hahn, B. Zengler, S. Queins „UML 2 glasklar”. ISBN 3-446-22575-7, Carl Hanser Verlag, 2004
- [Josuttis, 2008] N. Josuttis „SOA in der Praxis – System-Design für verteilte Geschäftsprozesse”. dpunkt Verlag, 2008.
- [Kabadayi, 2006] S. Kabadayi, A. Pridgen, C. Julien „Virtual Sensors: Abstracting Data from Physical Sensors”. TR-UTEDGE-2006-001, The University of Texas at Austin, 2006

- [Kalfoglou, 2003] Y. Kalfoglou, M. Schorlemmer „Ontology mapping: the state of the art”. The Knowledge Engineering Review, Cambridge Journals Online, DOI: 10.1017/S0269888903000651, 2003
- [Katifori, 2007] A. Katifori, C. Halatsis, G. Lepouras, C. Vassilakis, E. Giannopoulou „Ontology Visualization Methods – A Survey”. ACM Computing Surveys (CSUR) Surveys Homepage archive, Volume 39, Issue 4, 2007
- [Keen, 1980] P. Keen „Decision support systems: a research perspective”. Center for Information Systems Research, Alfred P. Sloan School of Management, Cambridge, Mass., 1980
- [Killalea, 2016] T. Killalea „The Hidden Dividends of Microservices”. acmqueue May-June 2016, 2016
- [Klopfer, 2009] M. Klopfer, I. Simonis „SANY an open service architecture for sensor networks”. ISBN 978-3-00-028571-4, 2009
- [Knöpfel, 2006] A. Knöpfel, B. Gröne, P. Tabeling „Fundamental Modeling Concepts: Effective Communication of IT Systems”. ISBN: 978-0-470-02710-3, Wiley, 2006
- [Kong, 2016] Q. Kong, R. M. Allen, L. Schreier, Y.-W. Kwon „MyShake: A smartphone seismic network for earthquake early warning and beyond”. Science advances, Vol. 2, Issue 2, 2016
- [Kourtesis, 2009] D. Kourtesis, I. Paraskakis „Supporting Semantically Enhanced Web Service Discovery for Enterprise Application Integration”. In G. Mentzas, P. Gouvas, T. Bouras, A. Friesen (Eds.) „Semantic Enterprise Application Integration for Business Processes: Service-Oriented Frameworks”, Hersley: IGI Global, 2009.
- [Kreps, 2014] J. Kreps „Questioning the Lambda Architecture”. <http://radar.oreilly.com/2014/07/questioning-the-lambda-architecture.html>, letzter Abruf am 4.9.2016, 2014.
- [Krumeich, 2014] J. Krumeich, B. Weis, D. Werth, P. Loos „Event-Driven Business Process Management: where are we now?”. Business Process Management Journal, Vol. 20 Issue 4, S. 615 - 633, 2014
- [Lang, 1995] S. M. Lang, P. C. Lockemann „Datenbankeinsatz”. Springer-Verlag, 1995

- [Lefort, 2011] L. Lefort, C. Henson, K. Taylor (Eds.), „Semantic Sensor Network XG Final Report“. W3C Incubator Group Report, 2011
- [Liebhart, 2007] D. Liebhart „SOA goes real – Service-orientierte Architekturen erfolgreich planen und einführen“. Hanser Fachbuchverlag, ISBN 978-3446410886, 2007
- [Lindel, 2007] M. K. Lindel, C. Prater, R. W. Perry „Emergency Management“. John Wiley & Sons, New York, 2007
- [Liu, 2013] S. Liu, C. Brewster, D. Shaw „Ontologies for Crisis Management: A Review of State of the Art in Ontology Design and Usability“. Proceedings of the 10th International ISCRAM Conference, 2013
- [Lubos, 2007] M. Lubos, K. Jana „Presentation of environmental information to public with spatial information systems“. In Environmental Software Systems - Dimensions of Environmental Informatics, 7th International Symposium on Environmental Software Systems (ISESS 2007) S. 368, 2007
- [Luckham, 2012] D. Luckham „Event Processing for Business: Organizing the Real-Time Enterprise“. John Wiley & Sons, Hoboken, NJ, 2012
- [Mahnke, 2009] W. Mahnke, S. Leitner, M. Damm „OPC Unified Architecture“. Springer Verlag, ISBN: 978-3540688983, 2009
- [Maier, 1998] M. W. Maier „Architecting principles for systems-of-systems“. Systems Engineering, Volume 1, Issue 4, 1998
- [Mayr, 2009] R. Mayr, C. Sierpinski, J. Oehmen „Frühwarnung als Produkt-Lebensretter“. Sicherheit & Industrie, März 2009
- [MacKenzie, 2006] C. M. MacKenzie, K. Laskey, F. McCabe, P. F. Brown, R. Metz (Ed.) „Reference Model for Service Oriented Architecture 1.0“. OASIS, 2006.
- [Malizia, 2010] A. Malizia, T. Onorati, P. Diaz, I. Aedo, F. Astorga-Paliza „SEMA4A: An ontology for emergency notification systems accessibility“. Expert Systems with Applications, Volume 37, Issue 4, S. 3380–3391, 2010

- [Marklund, 2016] J. Marklund, C. H. Wiklund „Learning from C2 Situations in the Field – Identifying Lessons from a Major Forest Fire in Sweden”. Proceedings of the ISCRAM 2016 Conference, 2016
- [Marz, 2015] N. Marz, J. Warren „Big Data: Principles and Best Practices of Scalable Realtime Data Systems”. Manning Publications Co., Greenwich, CT, USA, 2015
- [Meissen, 2008] U. Meissen, A. Voisard „Increasing the effectiveness of early warning via context-aware alerting”. Proceedings of 5th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2008), S. 431-440, 2008
- [Meissen, 2010] U. Meissen, A. Voisard „Towards a Reference Architecture for Early Warning Systems”. Proceedings of International Conference on Intelligent Networking and Collaborative Systems, IEEE, DOI: 10.1109/INCOS.2010.81, 2010
- [Meissen, 2013] U. Meissen, D. Faust, F. Fuchs-Kittowski „WIND – A meteorological early warning system and its extensions towards mobile services”. Proceedings of EnviroInfo 2013: Environmental Informatics and Renewable Energies, ISBN: 978-3-8440-1676-5, 2013
- [Meissen, 2014a] U. Meissen, F. Fuchs-Kittowski „Towards a reference architecture of crowdsourcing integration in early warning systems”. Proceedings of the 11th International ISCRAM Conference, University Park, Pennsylvania, USA, 2014
- [Meissen, 2014b] U. Meissen, M. Hardt, A. Voisard „Towards a general system design for community-centered crisis and emergency warning systems”. Proceedings of the 11th International ISCRAM Conference, University Park, Pennsylvania, USA, 2014
- [Michelson, 2006] B. M. Michelson „Event-Driven Architecture Overview”. DOI: 10.1571/bda2-2-06cc, 2006
- [Middleton, 2010] S. E. Middleton (Ed.) „SANY Fusion and Modelling Architecture”. Open Geospatial Consortium Inc., OGC® Discussion Paper, 2010
- [Mongula, 2009] D. B. Mongula „Enterprise Ontology of Flood Control Domain Master”. Delft University of Technology, 2009

- [Montanari, 2007] M. Montanari, S. Mehrotra, N. Venkatasubramanian „Architecture for an Automatic Customized Warning System”. Intelligence and Security Informatics, IEEE, S. 32-39, 2007
- [Nativi, 2015] S. Nativi, B. Domenico „OGC CF-netCDF 3.0 encoding using GML Coverage Application Schema”. Implementation Standard published by Open Geospatial Consortium (OGC), 2015
- [Neal, 1997] D. Neal „Reconsidering the Phases of Disaster”. International Journal of Mass Emergencies and Disasters, Vol. 15, No. 2, 1997
- [Negahban, 2016] M. Negahban, R. Nourjou „Internet of Things for Next-Generation Public Safety Mobile Communications”. Proceedings of the ISCRAM 2016 Conference, 2016
- [NEHRP, 2016] National Earthquake Hazards Reduction Program (NEHRP) „Introduction to emergency management”. <http://training.fema.gov/EMIWeb/EarthQuake/NEH0101220.htm>, letzter Abruf am 5.5.2016, 2016
- [Newman, 2015] S. Newman „Building Microservices”. O'Reilly Media, Inc., ISBN 1491950315, 2015
- [NGA, 1979] National Governors' Association Center for Policy Research „Comprehensive Emergency Management: A Governor's Guide”. Washington, 1979
- [Noy, 2000] N. F. Noy, R. W. Fergerson, M. A. Musen „The knowledge model of Protege-2000: Combining interoperability and flexibility”. In Proceedings of 2nd International Conference on Knowledge Engineering and Knowledge Management (EKAW'2000), Juanles-Pins, France, 2000
- [Oerter, 1968] G. W. Oerter „A New Implementation of Decision Tables for a Process Control Language”. IEEE Transactions on Industrial Electronics and Control Instrumentation, Vol. IECI-15, No. 2, Dezember, 1968
- [OMG, 2013] Object Management Group (OMG) „Business Process Model and Notation (BPMN) - Version 2.0.2”.
<http://www.omg.org/spec/BPMN/2.0.2/>, letzter Abruf am 4.9.2016, 2013

- [OMG, 2016] Object Management Group (OMG) „Decision Model and Notation (DMN) V1.1”. <http://www.omg.org/spec/DMN/1.1>, letzter Abruf am 4.9.2016, 2016
- [Onpulson, 2016] Onpulson „Onpulson Wirtschaftslexikon”. Campus-Verlag, <http://www.onpulson.de/lexikon/>, letzter Abruf am 5.5.2016, 2016
- [OpenGroup, 1995] The Open Group „The Open Group Architecture Framework (TOGAF)”. Van Haren Publishing, ISBN: 9789087536794, 1995
- [ORACLE, 2012] ORACLE „De-mystifying eventual *consistency* in distributed systems”. ORACLE NOSQL DATABASE Dokumentation, Juni, 2012
- [Papazoglou, 2004] M. P. Papazoglou, D. Georgakopoulos „Service-oriented computing”. Communications of the ACM 46(10), S. 24-28, 2004
- [Patig, 2010] S. Patig, V. Casanova-Brito, B. Vögeli: „IT Requirements of Business Process Management in Practice - An Empirical Investigation”. Proceedings of the 8th International Conference on Business Process Management (BPM 2010), Springer, 2010
- [Pengel, 2013] B. E. Pengel, V. V. Krzhizhanovskaya, N. B. Melnikova, G. S. Shirshov, A. R. Koelewijn, A. L. Pyayt, I. I. Mokhov „Flood Early Warning System: Sensors and Internet”. Floods: From Risk to Opportunity, IAHS Red Book N 357, IAHS Publ. 357, S. 445-453, 2013
- [Pettway, 1980] R. F. Pettway, J. F. Sinkey Jr. „Establishing On-Site Bank Examination Priorities: An Early-Warning System Using Accounting and Market Information”. The Journal of Finance, Volume 35, Issue 1, 1980
- [Pietzuch, 2003] P. R. Pietzuch, S. Bhola „Congestion Control in a Reliable Scalable Message-Oriented Middleware”. Proceedings of the ACM/IFIP/USENIX 2003 International Conference on Middleware, S. 202-221, 2003
- [Pinto, 2014] D. Pinto, A. Cipriano „Tsunami Early Warning System Architecture based on Automation Technologies”. 1st International Conference on Information and Communication Technologies for Disaster Management (ICT-DM), IEEE, DOI: 10.1109/ICT-DM.2014.6917785, 2014

- [Pohl, 2008] K. Pohl „Requirements Engineering. Grundlagen, Prinzipien, Techniken“. Dpunkt, Heidelberg, 2008
- [Power, 2002] D. J. Power „Categorizing decision support systems: a multi-dimensional approach“. In Decision making support systems, Manuel Mora, Guisseppi A. Forgionne, and Jatinder N. D. Gupta. IGI Publishing, Hershey, PA, USA. ISBN 1-59140-045-7, 2002
- [Preist, 2004] C. Preist „A conceptual architecture for semantic web services“. In McIlraith, S.A., Plexousakis, D. and van Harmelen, F. (Ed.): The Semantic Web – ISWC 2004, LNCS Volume 3298. Springer-Verlag Berlin Heidelberg, S. 395-409, 2004
- [Putman, 2001] J. R. Putman „Architecting with RM-ODP“. Prentice Hall, ISBN 0-13-019116-7, 2001
- [Rainieri, 2010] C. Rainieri, G. Fabbrocino, E. Cosenza „Integrated seismic early warning and structural health monitoring of critical civil infrastructures in seismically prone areas“. Structural Health Monitoring, DOI: 10.1177/1475921710373296, 2010
- [Ramar, 2012] K. Ramar, T. T. Mirnalinee „An Ontological Representation for Tsunami Early Warning System“. Proceedings of IEEE-International Conference on Advances in Engineering, Science and Management (ICAESM 2012), 2012
- [Reiner, 1991] U. Reiner „Anfragesprachen für Informationssysteme“. Reihe Informationswissenschaft der DGD, Bd. I, Deutsche Gesellschaft für Dokumentation, 1991
- [Reiter, 2000] E. Reiter, R. Dale „Building Natural Language Generation Systems“. Cambridge University Press, ISBN 0521620368, 2000
- [Richling, 2003] J. Richling „Beispielarchitekturen I“. Vorlesung Wintersemester 2003/2004, Humboldt-Universität Berlin, 2003
- [Richling, 2011] J. Richling, M. Werner, M. C. Jaeger, G. Mühl, H. Heiß „Autonomie in verteilten IT-Architekturen“. Oldenbourg Verlag, 2011
- [Robinson, 2015] J. J. Robinson, J. Maddock, K. Starbird Examining the Role of Human and Technical Infrastructure during Emergency Response. Proceedings of the ISCRAM 2015 Conference, 2015

- [Rücker, 2016] B. Rücker „Der Brückenschlag - Warum DMN den Business-Rules-Engines-Markt aufmischt“. Business Technology 01/2016, 2016
- [Rudloff, 2006] A. Rudloff, E. R. Flueh, W. Hanka, J. Lauterjung, T. Schöne „The German-Indonesian Tsunami Early Warning System (GITEWS)“. 3rd General Assembly European Geosciences Union, Vienna, Austria, 2006
- [Rupp, 2007] C. Rupp „Requirements-Engineering und -Management: Professionelle, iterative Anforderungsanalyse für die Praxis“. 4. Auflage, Carl Hanser Verlag GmbH & Co. KG, 2007
- [Sakaki, 2010] T. Sakaki, M. Okazaki, Y. Matsuo „Earthquake Shakes Twitter Users - Real-time Event Detection by Social Sensors“. Proceedings of WWW2010, 26.-30. April, Raleigh, North Carolina, 2010
- [Sanderson, 2005] N. Sanderson, V. Goebel, E. Munthe-Kass „Metadata management for ad-hoc InfoWare - a rescue and emergency use case for mobile ad-hoc scenarios“. Proceedings of International Conference on Ontologies, Databases and Applications of Semantics (ODBASEOS), Cyprus, 2005
- [Satriano, 2011] C. Satriano, L. Elia, C. Martino, M. Lancieri, A. Zollo, G. Iannaccone „PRESto, the earthquake early warning system for Southern Italy: Concepts, capabilities and future perspectives“. Soil Dynamics and Earthquake Engineering 31, S. 137–153, Elsevier, 2011
- [Schekkerman, 2004] J. Schekkerman „How to survive in the Jungle of Enterprise Architecture Frameworks: Creating or Choosing an Enterprise Architecture Framework“. Trafford Publishing, 2004
- [Schleipen, 2012] M. Schleipen „Adaptivität und semantische Interoperabilität von Manufacturing Execution Systemen (MES)“. Karlsruher Schriften zur Anthropomatik Band 12, Dissertation, Karlsruher Institut für Technologie, Fakultät für Informatik, ISBN 978-3-86644-955-8, 2012
- [Scholte, 2014] K. Scholte, L. Rothkrantz „Personal warning system for vessels under bad weather conditions“. Proceedings of the 11th International ISCRAM Conference, 2014

- [Sicilia, 2014] M. Sicilia (Ed.) „Handbook of Metadata, Semantics and Ontologies”. World Scientific Pub Co Inc., ISBN: 978-9812836298, 2014
- [Sharma, 2016] Vinod Kumar Sharma, G. Srinivasa Rao, E. Amminedu, P. V. Nagamani, Abhinav Shukla, K. Ram Mohan Rao, V. Bhanumurthy „Event-driven flood management: design and computational modules”. Geo-spatial Information Science, 19:1, S. 39-55, DOI: 10.1080/10095020. 2016.1151212, 2016
- [Shi, 2015] H. Shi, T. Li, R. Liu, J. Chen, J. Li, A. Zhang, G. Wang „A service-oriented architecture for ensemble flood forecast from numerical weather prediction”. Journal of Hydrology 527, S. 933-942, 2015
- [Simonis, 2008] I. Simonis (Ed.) „OGC® Sensor Web Enablement Architecture - Best Practice”. Open Geospatial Consortium, Inc., 2008
- [Sommerville, 2012] I. Sommerville „Software Engineering”. Pearson, München, 2012
- [Sprague, 1980] R. Sprague „A Framework for the Development of Decision Support Systems”. MIS Quarterly. Vol. 4, No. 4, 1980
- [Staab, 2004] S. Staab, R. Studer „Handbook on Ontologies”. International Handbooks on Information Systems, Springer, 2004
- [Starke, 2008] G. Starke „Effektive Software-Architekturen - Ein praktischer Leitfaden”. Hanser Verlag 3. Auflage, ISBN-10 3-446-41215-8, 2008
- [Stuckenschmidt, 2009] H. Stuckenschmidt „Ontologien: Konzepte, Technologien und Anwendungen”. Informatik im Fokus, Springer-Verlag, 2009
- [Tanenbaum, 1985] A. S. Tanenbaum, R. van Renesse „Distributed Operating Systems”. ACM Computing Surveys, 17(4), 1985
- [Tanenbaum, 2006] A. Tanenbaum, M. van Steen „Distributed Systems - Principles and Paradigms”. Prentice Hall, 2. Auflage, 2006
- [Temnikova, 2015] I. Temnikova, C. Castillo, S. Vieweg „EMTerms 1.0: A Terminological Resource for Crisis Tweets”. Proceedings of the ISCRAM 2015 Conference, 2015

- [Thoma, 2014] K. Thoma „Resilien-Tech *Resilience-by-Design*: Strategie für die technologischen Zukunftsthemen“. acatech STUDIE April 2014, acatech - Deutsche Akademie der Technikwissenschaften, 2014
- [Thönes, 2015] J. Thönes "Microservices". IEEE Software January/February 2015, IEEE Computer Society, 2015
- [TOG, 2011] The Open Group „TOGAF Version 9.1 - The Open Group Architecture Framework (TOGAF)“. 2009
- [Trachsler, 2009] D. Trachsler „Resilienz: Konzept zur Krisen- und Katastrophenbewältigung“. CSS Analysen zur Sicherheitspolitik, ETH Zürich, September 2009
- [Turoff, 2002] M. Turoff „Past and Future Emergency Response Information Systems“. Communications of the ACM, 45, 4, S. 29-33, 2002
- [UNISDR, 2005] United Nations Office for Disaster Risk Reduction (UNISDR) „Hyogo Framework for Action (HFA)“. Geneva, Switzerland, 2005
- [UNISDR, 2006] UN/ISDR Platform for the Promotion of Early Warning (PPEW) „Developing Early Warning Systems: A Checklist“. Third International Conference on Early Warning (EWC III), Bonn, Germany, 2006
- [UNISDR, 2009] United Nations Office for Disaster Risk Reduction (UNISDR) „2009 UNISDR terminology on disaster risk reduction“. Geneva, Switzerland, 2009
- [UNISDR, 2015] United Nations Office for Disaster Risk Reduction (UNISDR) „Sendai Framework for Disaster Risk Reduction“. Geneva, Switzerland, 2015
- [Uschold, 1995] M. Uschold and M. King „Towards a Methodology for Building Ontologies“. AIAI-TR-183, Workshop on Basic Ontological Issues in Knowledge Sharing in conjunction with IJCAI-95, 1995
- [Usländer, 2007] T. Usländer „Reference Model for the ORCHESTRA Architecture (RM-OA) V2 (Rev 2.1)“. OGC® Best Practice, Open Geospatial Consortium Inc., 2007
- [Usländer, 2009] T. Usländer (Ed.) „Specification of the Sensor Service Architecture, Version 3.0 (Rev. 3.1)“. OGC Discussion Paper 09-132r1.

- Deliverable D2.3.4 of the European Integrated Project SANY, FP6-IST-033564, http://portal.opengeospatial.org/files/?artifact_id=35888, letzter Abruf am 4.9.2016, 2009
- [Usländer, 2010] T. Usländer „Service-Oriented Design of Environmental Information Systems“. Karlsruher Schriften zur Anthropomatik, KIT Scientific Publishing, ISBN: 978-3-86644-499-7, 2010
- [Valtonen, 2004] E. Valtonen, R. Addams-Moring, T. Virtanen, A. Järvinen, M. Moring „Emergency Announcements to Mobile User Devices in Geographically Defined Areas“. Proceedings ISCRAM 2004, S. 151–156, Brussels, Belgium, 2004
- [Van de Walle, 2008] B. Van de Walle, M. Turoff „Decision support for emergency situations“. Inf Syst E-Bus Manage, Springer, DOI: 10.1007/s10257-008-0087-z, 2008
- [W3C, 2016] W3C „LargeTripleStores“, <https://www.w3.org/wiki/LargeTripleStores>, letzter Abruf am 27.7.2016, 2016
- [Wächter, 2009] J. Wächter, M. Lendholt, M. Hammitzsch „Introduction to the Distant Early Warning System (DEWS)“. Proceedings of the DEWS Midterm Conference, 2009
- [Wächter, 2012] J. Wächter, A. Babeyko, J. Fleischer, R. Häner, M. Hammitzsch, A. Kloth, M. Lendholt „Development of tsunami early warning systems and future challenges“. Natural Hazards and Earth System Sciences 12, DOI: 10.5194/nhess-12-1923-2012, 2012
- [Wand, 1990] Y. Wand, R. Weber „An ontological model of an information system“. IEEE Transactions on Software Engineering, Band 16, 1990.
- [Wang, 2010] J. Wang, J. Bigham, J. Chew, M. Novkovic, I. Dattani „Adding Resilience to Message Oriented Middleware“. Serene 2010 ACM, 13.-16. April, London, UK, 2010
- [Wenzel, 2014] F. Wenzel, J. Zschau (Eds.) „Early Warning for Geological Disasters - Scientific Methods and Current Practice“. Springer Verlag, DOI: 10.1007/978-3-642-12233-0, 2014
- [Xiao, 2016] C. Xiao, N. Chen, X. Wang, Z. Chen „A Semantic Registry Method Using Sensor Metadata Ontology to Manage Heterogeneous Sensor

- Information in the Geospatial Sensor Web". ISPRS International Journal of Geo-Information 2016, 5, 63; DOI: 10.3390/ijgi5050063, 2016
- [Zachman, 1987] J. Zachman „A framework for information systems architecture". IBM Systems Journal 26, Nr. 3, S. 276–292, 1987
- [Zeit, 2014] Zeit Online „Tsunami 2004: Man hätte Stunden gehabt, Menschen zu warnen". <http://www.zeit.de/wissen/umwelt/2014-12/tsunami-indischer-ozean-flutwelle-jahrestag>, letzter Abruf am 7.5.2016, 2014
- [ZKI, 2013] Zentrum für Satellitengestützte Kriseninformation (ZKI), „Krisenkreislauf". <https://www.zki.dlr.de/de/image/938>, Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR) , letzter Abruf am 7.5.2016, 2013
- [Zhang, 2007] L.-J. Zhang, S. Cheng, Y.-M. Chee, A. Allam, Q. Zhou „Pattern Recognition based Adaptive Categorization Technique and Solution for Services Selection". Proceedings of IEEE Asia-Pacific Services Computing Conference, 2007.
- [Zielinski, 2013] A. Zielinski, S. E. Middleton, L. N. Tokarchuk, X. Wang „Social Media Text Mining and Network Analysis for Decision Support in Natural Crisis Management". Proceedings of the 10th International ISCRAM Conference, 2013.
- [Zimmermann, 2003] T. Zimmermann, S. Diehl, A. Zeller „How History Justifies System Architecture (or not)". Proc. International Workshop on Principles of Software Evolution (IWPSE 2003), Helsinki, Finland, 2003

Eigene Veröffentlichungen (chronologisch)

- [Moßgraber, 1997] J. Moßgraber „Konzeption, Entwurf und Umsetzung eines Metadatenmodells zur Interpretation und Verwaltung von Informationen mit geographischem Bezug“. Diplomarbeit, Fakultät für Informatik, Universität Karlsruhe, 1997
- [Gebhart, 2010] M. Gebhart, J. Moßgraber, T. Usländer, S. Abeck „SoaML-basierter Entwurf eines dienstorientierten Überwachungssystems“, Informatik 2010 (Leipzig) - Workshop „SOA und Standardsoftware - Standardsoftware wo möglich und Individualität wo nötig“, 2010
- [Moßgraber, 2010] J. Moßgraber, F. Reinert, H. Vagts „An Architecture for a Task-Oriented Surveillance System - A Service- and Event-Based Approach“. ICONS 2010, IEEE Computer Society Digital Library (CSDL), 2010
- [Wanner, 2010] L. Wanner, H. Bosch, N. Bouayad-Agha, U. Bügel, G. Casamayor, T. Ertl, A. Karppinen, I. Kompatsiaris, T. Koskentalo, S. Mille, J. Moßgraber, A. Moumtzidou, M. Myllynen, E. Pianta, M. Rospocher, H. Saggion, L. Serafini, V. Tarvainen, S. Tonelli, T. Usländer, S. Vrochidis „Service-Based Infrastructure for User-Oriented Environmental Information Delivery“. 24th International Conference on Informatics for Environmental Protection, Bonn, 2010
- [Bosch, 2011] H. Bosch, D. Thom, T. Ertl, L. Wanner, N. Bouayad-Agha, U. Bügel, G. Casamayor, A. Karppinen, I. Kompatsiaris, T. Koskentalo, S. Mille, J. Moßgraber, A. Moumtzidou, M. Myllynen, E. Pianta, M. Rospocher, H. Saggion, L. Serafini, V. Tarvainen, S. Tonelli, T. Usländer, S. Vrochidis „Das Web als personalisierte Entscheidungsplattform - Die PESCaDO Idee“. Informatik 2011, 41. Jahrestagung der Gesellschaft für Informatik, 4.-7.10.2011, TU Berlin, 2011
- [Wanner, 2011] L. Wanner, S. Tonelli, S. Vrochidis, J. Moßgraber, H. Bosch, A. Karppinen, M. Myllynen, M. Rospocher, N. Bouayad-Agha, U. Bügel, G. Casamayor, T. Ertl, I. Kompatsiaris, T. Koskentalo, S. Mille, A.

- Moumtzidou, E. Pianta, H. Saggion, L. Serafini, V. Tarvainen, T. Usländer „Building an Environmental Information System for Personalized Content Delivery”. International Symposium on Environmental Software Systems (ISESS), 2011
- [Bouayad-Agha, 2012] N. Bouayad-Agha, G. Casamayor, S. Mille, J. Moßgraber, M. Rospocher, H. Saggion, L. Serafini, L. Wanner „Generation of Multilingual Personalized Environmental Bulletins from an OWL-based Ontology”. EnviroInfo 2012, 29.-31.08.2012, Dessau, 2012
- [Moßgraber, 2012a] J. Moßgraber, S. Middleton, M. Hammitzsch, S. Poslad „A Distributed Architecture for Tsunami Early Warning and Collaborative Decision-support in Crises”. European Geosciences Union General Assembly (EGU) 2012, 22.–27. April 2012, Vienna, 2012
- [Moßgraber, 2012b] J. Moßgraber, M. Rospocher „Ontology Management in a Service-oriented Architecture”. DEXA (International Workshop on Web Semantics and Information Processing) 2012, 09/2012, Vienna, Austria, 2012
- [Moumtzidou, 2012] A. Moumtzidou, V. Epitropou, S. Vrochidis, S. Voth, A. Bassoukos, K. Karatzas, J. Moßgraber, I. Kompatsiaris, A. Karppinen, J. Kukkonen „Environmental Data Extraction from Multimedia Resources”. MAED 2012 - ACM International Workshop on Multimedia Analysis for Ecological Data (MAED 2012), November 2, Nara (Japan), 2012
- [Tao, 2012] Ran Tao, S. Poslad, J. Moßgraber, S. Middleton, M. Hammitzsch „Scalable and Resilient Middleware to Handle Information Exchange during Environment Crisis”. European Geosciences Union General Assembly (EGU) 2012, 22.–27. April 2012, Vienna, 2012
- [Vrochidis, 2012] S. Vrochidis, V. Epitropou, A. Bassoukos, S. Voth, K. Karatzas, A. Moumtzidou, J. Moßgraber, I. Kompatsiaris, A. Karppinen, J. Kukkonen „Extraction of Environmental Data from on-line Environmental Information Sources”. 3rd Intelligent Systems for Quality of Life information Services Workshop (ISQL 2012) - Part of 8th AIAI Conference, Volume 0382 of the IFIP Advances in Information and Communication Technology series, 2012

- [Wanner, 2012a] L. Wanner, S. Vrochidis, M. Rospocher, J. Moßgraber, H. Bosch, A. Karppinen, M. Myllynen, S. Tonelli, N. Bouayad-Agha, G. Casamayor, T. Ertl, D. Hilbring, L. Johansson, K. Karatzas, I. Kompatsiaris, T. Koskentalo, S. Mille, A. Moutmtzidou, E. Pianta, L. Serafini, V. Tarvainen „Personalized Environmental Service Orchestration for Quality Life Improvement”. 3rd Intelligent Systems for Quality of Life information Services Workshop (ISQL 2012) - Part of 8th AIAI Conference, Volume 0382 of the IFIP Advances in Information and Communication Technology series, 2012
- [Wanner, 2012b] L. Wanner, M. Rospocher, S. Vrochidis, H. Bosch, N. Bouayad-Agha, U. Bügel, G. Casamayor, T. Ertl, Desiree Hilbring, A. Karppinen, I. Kompatsiaris, T. Koskentalo, S. Mille, J. Moßgraber, A. Moutmtzidou, M. Myllynen, E. Pianta, H. Saggion, L. Serafini, V. Tarvainen, S. Tonelli „Personalized Environmental Service Configuration and Delivery Orchestration: The PESCaDO Demonstrator”. Extended Semantic Web Conference (ESWC) 2012, 27-31.5.2012, Heraklion Greece, 2012
- [Häner, 2013] R. Häner, J. Wächter, M. Hammitzsch, M. Lendholt, J. Moßgraber „TRIDEC System-of-Systems, Choreography of large-scale concurrent tasks in Natural Crisis Management”, IEEE Eleventh International Symposium on Autonomous Decentralized Systems (ISADS), 2013, Mexico City, Mexico, ISBN: 978-1-4673-5069-3, DOI: 10.1109/ISADS.2013.6513440, 2013
- [Hilbring, 2013] D. Hilbring, A. Moutmtzidou, J. Moßgraber, S. Vrochidis „Semantically Enriching an Open Source Sensor Observation Service Implementation for Accessing Heterogeneous Environmental Data Sources”. Transactions in GIS, 2013, DOI: 10.1111/tgis.12055, 2013
- [Chaves, 2013] F. Chaves, J. Moßgraber, M. Schenk, U. Bügel „Semantic Registries for Heterogeneous Sensor Networks - Bridging the semantic gap for collaborative crises management”. 24th International Workshop on Database and Expert Systems Applications (DEXA) 2013, August 26 - 29, 2013, Prague, Czech Republic, DOI: 10.1109/DEXA.2013.18, 2013
- [Löwe, 2013] P. Löwe, J. Wächter, M. Hammitzsch, M. Lendholdt, R. Häner, J. Moßgraber, Z. Sabeur „The Evolution of Disaster Early Warning Systems

in the TRIDEC Project". The International Society of Offshore and Polar Engineers (ISOPE) Conference 2013, Anchorage, Alaska, USA, June 30 to July 5, 2013

[Moßgraber, 2013] J. Moßgraber, F. Chaves, S. E. Middleton, Z. Zlatev, R. Tao „The Seven Main Challenges of an Early Warning System Architecture". ISCRAM 2013, Baden-Baden, Germany, 2013

[Moumtzidou, 2013] A. Moumtzidou, V. Epitropou, S. Vrochidis, K. Karatzas, S. Voth, A. Bassoukos, J. Moßgraber, A. Karppinen, J. Kukkonen and I. Kompatsiaris „A Model for Environmental Data Extraction from Multimedia and its Evaluation against various Chemical Weather Forecasting Datasets". Ecological Informatics, DOI: <http://dx.doi.org/10.1016/j.ecoinf.2013.08.003>, 2013

[Krempel, 2014] E. Krempel, P. Birnstill, Y. Fischer, E. Monari, J. Moßgraber, M. Schenk, J. Beyerer, G. Unmüßig „Privacy-aware Smart Video Surveillance Revisited". Future Security 2014, Berlin, 2014

[Moßgraber, 2014] J. Moßgraber, T. Kubera, A. Werner „More safety for football events: Improving the communication of stakeholders and the dialogue with fans". Future Security 2014, Berlin, 2014

[Moßgraber, 2014] J. Moßgraber, D. Hilbring „Automating the web publishing process of environmental data by using semantic annotations". International Workshop on Environmental Multimedia Retrieval '14 in conjunction with the ACM Conference on Multimedia Retrieval '14, Glasgow UK, 2014

[Bernard, 2015a] T. Bernard, J. Moßgraber, A. E. Madar, A. Rosenberg, J. Deuerlein, H. Lucas, K. Boudergui, D. Ilver, E. Brill, N. Ulitzur „Realtime Detection and Mitigation of CBRN related Contamination Events of Drinking Water". Future Security 2015, September 15-17, Berlin, 2015

[Bernard, 2015b] T. Bernard, J. Moßgraber, A. E. Madar, A. Rosenberg, J. Deuerlein, H. Lucas, K. Boudergui, D. Ilver, E. Brill, N. Ulitzur „SAFEWATER - Innovative Tools for the Detection and Mitigation of CBRN related Contamination Events of Drinking Water". Computing and Control for the Water Industry conference 2015 (CCWI2015), September 2-4, 2015, Leicester, UK

- [Moßgraber, 2015] J. Moßgraber, M. Schenk, D. Hilbring „Modelling of an Ontology for a Communication Platform - More safety at football events by improving the communication between stakeholders”. The Ninth International Conference on Advances in Semantic Processing (SEMAPRO) 2015, July 19-24, Nice, France, 2015
- [Pagel, 2015] F. Pagel, J. Moßgraber, C. Decker, J.-P. Germann „A Multi-Sensor Technology for Improving Security at Complex Scenarios with Increased Risk of Violence”. Future Security 2015, September 15-17, Berlin, 2015
- [Santamaria, 2015] E. Santamaria, J. Moßgraber, E. Brill, I. M. Arango „A System Architecture for the Detection and Mitigation of CBRN related Contamination Events of Drinking Water”. Computing and Control for the Water Industry conference 2015 (CCWI2015), September 2-4, Leicester, UK, <http://dx.doi.org/10.1016/j.proeng.2015.08.891>, 2015
- [Wanner, 2015] L. Wanner, H. Bosch, N. Bouayad-Agha, G. Casamayor, D. Hilbring, L. Johansson, K. Karatzas, A. Karppinen, T. Koskentalo, S. Mille, J. Moßgraber, A. Moumtzidou, M. Myllynen, E. Pianta, M. Rospocher, L. Serafini, V. Tarvainen, S. Tonelli, S. Vrochidis, I. Kompatsiaris, T. Ertl „Getting the environmental information across: From the web to the user”. Expert Systems, Volume 32, Issue 3, 1 June 2015, Pages 405-432, DOI: 10.1111/exsy.12100, 2015
- [Hilbring, 2016] D. Hilbring, J. Moßgraber, M. Schenk, J. Link „Twitter Sentiment Analysis for German Football Matches”. Proceedings of Future Security 2016, Berlin, Germany, 2016

Karlsruher Schriftenreihe zur Anthropomatik (ISSN 1863-6489)

Herausgeber: Prof. Dr.-Ing. Jürgen Beyerer

Die Bände sind unter www.ksp.kit.edu als PDF frei verfügbar
oder als Druckausgabe bestellbar.

- Band 1** Jürgen Geisler
Leistung des Menschen am Bildschirmarbeitsplatz. 2006
ISBN 3-86644-070-7

- Band 2** Elisabeth Peinsipp-Byma
**Leistungserhöhung durch Assistenz in interaktiven Systemen
zur Szenenanalyse.** 2007
ISBN 978-3-86644-149-1

- Band 3** Jürgen Geisler, Jürgen Beyerer (Hrsg.)
Mensch-Maschine-Systeme. 2010
ISBN 978-3-86644-457-7

- Band 4** Jürgen Beyerer, Marco Huber (Hrsg.)
**Proceedings of the 2009 Joint Workshop of Fraunhofer IOSB and
Institute for Anthropomatics, Vision and Fusion Laboratory.** 2010
ISBN 978-3-86644-469-0

- Band 5** Thomas Usländer
Service-oriented design of environmental information systems. 2010
ISBN 978-3-86644-499-7

- Band 6** Giulio Milighetti
**Multisensorielle diskret-kontinuierliche Überwachung und
Regelung humanoider Roboter.** 2010
ISBN 978-3-86644-568-0

- Band 7** Jürgen Beyerer, Marco Huber (Hrsg.)
**Proceedings of the 2010 Joint Workshop of Fraunhofer IOSB and
Institute for Anthropomatics, Vision and Fusion Laboratory.** 2011
ISBN 978-3-86644-609-0

- Band 8** Eduardo Monari
**Dynamische Sensorselektion zur auftragsorientierten
Objektverfolgung in Kameranetzwerken.** 2011
ISBN 978-3-86644-729-5

- Band 9** Thomas Bader
Multimodale Interaktion in Multi-Display-Umgebungen. 2011
ISBN 3-86644-760-8
- Band 10** Christian Frese
Planung kooperativer Fahrmanöver für kognitive Automobile. 2012
ISBN 978-3-86644-798-1
- Band 11** Jürgen Beyerer, Alexey Pak (Hrsg.)
Proceedings of the 2011 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory. 2012
ISBN 978-3-86644-855-1
- Band 12** Miriam Schleipen
Adaptivität und Interoperabilität von Manufacturing Execution Systemen (MES). 2013
ISBN 978-3-86644-955-8
- Band 13** Jürgen Beyerer, Alexey Pak (Hrsg.)
Proceedings of the 2012 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory. 2013
ISBN 978-3-86644-988-6
- Band 14** Hauke-Hendrik Vagts
Privatheit und Datenschutz in der intelligenten Überwachung: Ein datenschutzgewährendes System, entworfen nach dem „Privacy by Design“ Prinzip. 2013
ISBN 978-3-7315-0041-4
- Band 15** Christian Kühnert
Data-driven Methods for Fault Localization in Process Technology. 2013
ISBN 978-3-7315-0098-8
- Band 16** Alexander Bauer
Probabilistische Szenenmodelle für die Luftbildauswertung. 2014
ISBN 978-3-7315-0167-1
- Band 17** Jürgen Beyerer, Alexey Pak (Hrsg.)
Proceedings of the 2013 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory. 2014
ISBN 978-3-7315-0212-8
- Band 18** Michael Teutsch
Moving Object Detection and Segmentation for Remote Aerial Video Surveillance. 2015
ISBN 978-3-7315-0320-0

- Band 19** Marco Huber
Nonlinear Gaussian Filtering: Theory, Algorithms, and Applications. 2015
ISBN 978-3-7315-0338-5
- Band 20** Jürgen Beyerer, Alexey Pak (Hrsg.)
Proceedings of the 2014 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory. 2014
ISBN 978-3-7315-0401-6
- Band 21** Todor Dimitrov
Permanente Optimierung dynamischer Probleme der Fertigungssteuerung unter Einbeziehung von Benutzerinteraktionen. 2015
ISBN 978-3-7315-0426-9
- Band 22** Benjamin Kühn
Interessengetriebene audiovisuelle Szenenexploration. 2016
ISBN 978-3-7315-0457-3
- Band 23** Yvonne Fischer
Wissensbasierte probabilistische Modellierung für die Situationsanalyse am Beispiel der maritimen Überwachung. 2016
ISBN 978-3-7315-0460-3
- Band 24** Jürgen Beyerer, Alexey Pak (Hrsg.)
Proceedings of the 2015 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory. 2016
ISBN 978-3-7315-0519-8
- Band 25** Pascal Birnstill
Privacy-Respecting Smart Video Surveillance Based on Usage Control Enforcement. 2016
ISBN 978-3-7315-0538-9
- Band 26** Philipp Woock
Umgebungskartenschätzung aus Sidescan-Sonardaten für ein autonomes Unterwasserfahrzeug. 2016
ISBN 978-3-7315-0541-9
- Band 27** Janko Petereit
Adaptive State \times Time Lattices: A Contribution to Mobile Robot Motion Planning in Unstructured Dynamic Environments. 2017
ISBN 978-3-7315-0580-8

Band 28 Erik Ludwig Krempel
**Steigerung der Akzeptanz von intelligenter Videoüberwachung
in öffentlichen Räumen.** 2017
ISBN 978-3-7315-0598-3

Band 29 Jürgen Moßgraber
Ein Rahmenwerk für die Architektur von Frühwarnsystemen. 2017
ISBN 978-3-7315-0638-6

Lehrstuhl für Interaktive Echtzeitsysteme
Karlsruher Institut für Technologie

Fraunhofer-Institut für Optronik, Systemtechnik und
Bildauswertung IOSB Karlsruhe

Frühwarnsysteme dienen zur möglichst frühzeitigen Information über eine sich anbahnende oder auftretende Gefahr, um Personen und Organisationen die Möglichkeit zu geben entsprechend darauf reagieren zu können. Um entscheiden zu können ob eine echte Gefährdung vorliegt, müssen alle relevanten Informationen den jeweiligen Entscheidern zur Verfügung stehen. Ein entsprechendes Unterstützungssystem muss hierzu Daten aggregieren, aufbereiten und in übersichtlicher Form präsentieren.

Die Konzeption eines Frühwarnsystems stellt komplexe Herausforderungen an die Systemarchitekten, da sie mit der gesamten Bandbreite von Problemen der aktuellen Informatikforschung konfrontiert werden, wie zum Beispiel der Analyse großer Datenmengen, dem Internet der Dinge oder der Informationsgewinnung aus sozialen Netzwerken.

Um diese Aufgabe zu erleichtern definiert die vorliegende Arbeit ein Rahmenwerk für die Architektur zukünftiger Frühwarnsysteme. Dieses legt besonderen Augenmerk auf die Lösung verschiedenster Architekturprobleme mittels semantischer Technologien und der Automation von Arbeitsabläufen durch die Nutzung von Methoden zur Geschäftsprozessmodellierung.

ISSN 1863-6489
ISBN 978-3-7315-0638-6

