DISSERTATIONS IN
FORESTRY AND
NATURAL SCIENCES

Markus Stocker

Situation Awareness in Environmental Monitoring

Publications of the University of Eastern Finland
Dissertations in Forestry and Natural Sciences No 192

UNIVERSITY OF
EASTERN FINLAND

MARKUS STOCKER

Situation Awareness in Environmental Monitoring

Concepts, Implementation, Applications

Publications of the University of Eastern Finland
Dissertations in Forestry and Natural Sciences
No 192

Academic Dissertation
To be presented by permission of the Faculty of Science and Forestry for public examination in the Auditorium SN 200, Snellmania Building at the University of Eastern Finland, Kuopio, on November 27, 2015 at 12 o'clock noon

Department of Environmental Science

Grano Oy
Jyväskylä, 2015
Editor: Prof. Pertti Pasanen

Distribution:
University of Eastern Finland Library / Sales of publications
julkaisumyynti@uef.fi
http://www.uef.fi/kirjasto

Front cover picture:
SMEAR II, Hyytiälä, Finland

ISBN: 978-952-61-1907-6 (Print)
ISBN: 978-952-61-1908-3 (PDF)
ISSN: 1798-5668 (Print)
ISSN: 1798-5676 (PDF)
ISSNL: 1798-5668

Author Markus Stocker, M.Sc.
University of Eastern Finland
Department of Environmental Science
Yliopistonranta 1 E, Snellmania
P.O. Box 1627, 70211 Kuopio, Finland
markus.stocker@uef.fi

Supervisors Professor Mikko Kolehmainen, D.Sc. (Tech.)
University of Eastern Finland
Department of Environmental Science
Yliopistonranta 1 E, Snellmania
P.O. Box 1627, 70211 Kuopio, Finland
mikko.kolehmainen@uef.fi

Docent Mauno Rönkkö, Ph.D.
University of Eastern Finland
Department of Environmental Science
Yliopistonranta 1 E, Snellmania
P.O. Box 1627, 70211 Kuopio, Finland
mauno.ronkko@uef.fi

Reviewers Professor Peter Fox, Ph.D.
Rensselaer Polytechnic Institute
Climate Variability and Solar-Terrestrial Physics
Jonsson-Rowland Science Center, 1W19
110 8th Street, Troy, NY 12180, USA
pfox@cs.rpi.edu

Associate Professor Krzysztof Janowicz, Ph.D.
University of California, Santa Barbara
Geography Department
4830 Ellison Hall, Santa Barbara, CA 93106-4060, USA
jano@geog.ucsb.edu

Opponent Werner Leo Kutsch, Dr. habil.
Director General
Integrated Carbon Observation System (ICOS)
University of Helsinki
Department of Physics
Erik Palménin aukio 1
P.O. Box 48, 00014 Helsinki, Finland
werner.kutsch@icos-ri.eu

Abstract

Data obtained in environmental monitoring contribute to advancing our understanding of natural and human-made systems, and phenomena. Phenomena can be monitored by means of various techniques. Some techniques use environmental sensor networks to automate monitoring. Common to such networks is the often large amount of heterogeneous data that result from their operation. The acquisition, curation, access, and processing of such data are widely recognized research problems. Beyond data, there are also important research questions related to the broad issue of how to interpret data, in other words how to make sense of data.

In this dissertation, phenomena are objects in situations, i.e. objects in structured parts of reality observed by environmental monitoring systems. We suggest that environmental monitoring systems can utilize computational models to acquire knowledge about situations from data, and utilize technologies in knowledge representation and reasoning to support the curation and processing of acquired situational knowledge. Situational knowledge acquisition and processing may be automated and may thus be achieved in near real-time. Automation enables the technical components of environmental monitoring systems to obtain and maintain situation awareness.

We study theories, methods, and technologies relevant to the acquisition of situational knowledge from data and to the representation of situational knowledge. Using the identified theories, methods, and technologies we develop, implement in software, and validate on case studies a software process for the acquisition, curation, access, and processing of situational knowledge in situation-aware environmental monitoring systems.

The main contributions include an architecture for a software framework designed to support the implementation of situation-aware environmental monitoring systems. Second, we propose an open source implementation for the architecture. Third, using the implementation, we develop concrete applications and discuss the acquisition, curation, access, and processing of situational knowledge in case studies with environmental monitoring in intelligent transportation systems, atmospheric science, and agricultural science.

The software framework architecture is aligned with a reference model for environmental research infrastructure. The reference model highlights that state of the art environmental research infrastructures are predominantly data-based systems. As the fourth contribution, we propose to extend the reference model with functionality for knowledge acquisition from data, and knowledge curation, access, and processing. The resulting model advances knowledge-based environmental research infrastructure.

A major strength of the proposed software framework is its support for both inductive data-driven and deductive knowledge-driven techniques in situational knowledge acquisition and processing. The presented applications suggest that the hybrid approach enables the development of situation-aware environmental monitoring systems with non-trivial situational knowledge acquisition problems. For applications in scientific research, e.g. in atmospheric science, the systems are, specifically, situation-aware environmental research infrastructure.

Universal Decimal Classification: 004.41, 004.62, 004.774.2, 004.8, 502.175
Library of Congress Subject Headings: Environmental monitoring; Situational awareness; Ontologies (Information retrieval); Information technology; Cyberinfrastructure; Expert systems (Computer science); Decision support systems; Software frameworks; Open source software; Semantic Web; Traffic monitoring; Atmospheric physics; Agriculture; Plant diseases
Yleinen suomalainen asiasanasto: ympäristö; ympäristön tila; monitorointi; tilannekuva; ontologiat; tietotekniikka; asiantuntijajärjestelmät; päätöksentukijärjestelmät; seuranta; tietokoneohjelmat; järjestelmäarkkitehtuuri; avoin lähdekoodi; semanttinen web

Preface

The dissertation marks the end of a period in my life that, up to the final year of graduate school, was improbable. Ask my high school teachers and I am certain some would argue *highly* improbable. Beyond a little inspiration and luck, of which there was more than a little, it is largely perseverance for which I am grateful, as it led to where I stand through a journey that was exciting as much as it was real.

Foremost, I want to express my gratitude to Prof. Mikko Kolehmainen, who made the journey possible. Not only did he adopt me with ease to his interdisciplinary research group, he also cared throughout the years that I was involved in interesting projects, which relieved me from having to worry about grants. However, it is the freedom I was given to explore ideas, fail and start over that I have come to appreciate most.

In fact, before I had the one that eventually became this dissertation, I more or less briefly contemplated various ideas including an ontology for soil types; study how information about consumption of, e.g., energy or water changes people's behaviour, if at all; and study the suitability of data visualization techniques, inspired by software visualization research, for the discovery of patterns in the environment. They all led nowhere.

Indeed, together with Docent Mauno Rönkkö, I was given so much freedom by my supervisors that I occasionally wondered whether it ought to be like that. Thank you Mauno for helping me to align dissertation and project work, a necessity without which my overtime would have gone through the roof. Thank you also for your positive attitude toward ideas and results.

I am grateful for the short discussions that sparked longer stretches during which we developed, implemented, and tested ideas. I would like to thank, in particular, Elham Baranizadeh, Dr. Ferdinando Villa, Emer. Prof. Pertti Martikainen, Dr. Harri Niska, Dr. Narasinha Shurpali, Hanna Huitu, Dr. Jussi Nikander, Dr. Christina Biasi, Eeva Lehtonen, Dr. Hannele Korhonen, Dr. Mika Komppula, and Paula Silvonen. The discussions and collaborations with you shaped this dissertation.

I like to acknowledge the University of Eastern Finland, its departments of Environmental Science and Applied Physics and the groups of Environmental Informatics and Aerosol Physics, the Department of Information Technologies at Åbo Akademi, the Department of Computer Science and Engineering at Aalto University, the Finnish Meteorological Institute, the Natural Resources Institute Finland, VTT Technical Research Center of Finland, Vaisala Oyj, Tekes, Academy of Finland, Profium Oy, HiQ Finland Oy, and Complexible Inc.

Let me express my gratitude to the person who invented white boards, and the person who installed the one at SN 3009/2, my office, as well as the white board itself. I spent many weekends drawing arrows and boxes, scribbling text, and erasing everything on that board. It is also fair to say that I realized what my dissertation will be while looking at that board and what I had scribbled on it on August 27, 2011. A board on a wall can be your friend.

I am grateful for the people I met and got to know beyond the office walls. Many contributed to holding me from running over the work-life balance cliff. Thank you Karin and Risto Koivisto for keeping me up-to-date with UEF news and for having been fabulous hosts and company over the years. Thank you Galina and Dr. Thomas Wirth for the many hours discussing all-matters-related-to-life over lunch and formidable dinners; Hristo Zlatev, Dr. Haritha Samaranayake, Dr. Hanna Stedt, Dr. Petra Korpisalo, Dr. Jarmo Laitinen, Prof. Jorma Palvimo, Dr. Ville Kolehmainen, Dr. Ale Närvänen for joining the discussions; Dr. Agnieszka Pacholska and Dr. Renata Gomes for having been part of them. Thank you Emer. Prof. Ossi Lindqvist for your continued effort to inspire others, including students in your Philosophy and History of Science lectures, and for all your links to, and hard copies of, articles you shared with me. Thank you Emer. Prof. Pertti Martikainen and your team for holding some of the finest courses I ever attended. Thank you Gregory Suszko for being passionate about music, for your work toward a University Orchestra, and for your help with proofreading manuscripts. Thanks to all board members for joining the leadership of the UEF Doctoral Student Association during 2014.

Prof. Abraham Bernstein (Avi) supervised my diploma dissertation at the University of Zurich and was the first person, ever, to motivate me for post-graduate studies. Thank you Avi for seeding the vision for the improbable journey that eventually became real, and done. Thank you Dr. Bijan Parsia for mentioning the interdisciplinary road: it has been inspiring so far.

Finally, I like to express my gratitude to Prof. Peter Fox and Assoc. Prof. Krzysztof Janowicz for their overall very positive appraisal of the dissertation manuscript, their time and valuable comments. Thank you Dr. habil. Werner Kutsch for being my opponent at the *viva voce*. I am also grateful for the comments by the anonymous reviewers of our publications as well as for the work of the global community of open source software developers.

The dissertation marks the end of a relatively short journey embedded in a longer life journey. I am exceptionally grateful for my family being part of it, for their continued unconditional love, shared also over great distances. I am thrilled to have met you, *azizam* Fatemeh Rostami, along the short journey, and at the prospect of you being part of the longer journey, and us one day smiling at the distant memories for where it all began.

Kuopio, October 21, 2015 *Markus Stocker*

Publications

The dissertation is based on the following publications:

I Stocker, M., Silvonen, P., Rönkkö, M., and Kolehmainen, M. (2015). Detection and classification of vehicles by measurement of road-pavement vibration and by means of supervised machine learning. *Journal of Intelligent Transportation Systems*. (In Press)
doi:10.1080/15472450.2015.1004063

II Stocker, M., Rönkkö, M., and Kolehmainen, M. (2014). Situational knowledge representation for traffic observed by a pavement vibration sensor network. *IEEE Transactions on Intelligent Transportation Systems*, 15(4):1441-1450.
doi:10.1109/TITS.2013.2296697

III Stocker, M., Baranizadeh, E., Portin, H., Komppula, M., Rönkkö, M., Hamed, A., Virtanen, A., Lehtinen, K., Laaksonen, A., and Kolehmainen, M. (2014). Representing situational knowledge acquired from sensor data for atmospheric phenomena. *Environmental Modelling & Software*, 58:27-47.
doi:10.1016/j.envsoft.2014.04.006

IV Stocker, M., Nikander, J., Huitu, H., Jalli, M., Koistinen, M., Rönkkö, M., and Kolehmainen, M. (2015). Plant disease pressure situation modelling in agriculture. *Environmental Modelling & Software*. (Submitted)

V Stocker, M., Rönkkö, M., and Kolehmainen, M. (2015). Knowledge-based environmental research infrastructure: moving beyond data. *Earth Science Informatics*. (In Press)
doi:10.1007/s12145-015-0230-6

Throughout this document, we refer to these papers using Roman numerals.

Contribution

The selected publications are peer-reviewed original research papers submitted, in press, or published in journals. This statement details author contribution toward each paper.

Paper **I** originated in project work pursued by the author on the detection and classification of vehicles using road-pavement vibration sensor data. This project was funded by Tekes, the Finnish Funding Agency for Innovation. The aim of the work was defined by the project, which was specified prior to my joining the research group in 2009. The author developed the system, performed the experiments, evaluated the results, and wrote the manuscript. Paula Silvonen contributed to performing the experiments and evaluating the results. The co-authors Paula Silvonen, Mauno Rönkkö, and Mikko Kolehmainen provided comments to the manuscript.

Paper **II** extends the approach pursued in Paper **I** with semantic technologies. It thus realized a first implementation of the core ideas underlying the dissertation. The author developed the system, performed the experiments, evaluated the results, and wrote the manuscript. The co-authors Mauno Rönkkö and Mikko Kolehmainen provided comments to the manuscript.

Paper **III** originated in discussions with Elham Baranizadeh. Elham addressed most of the domain specific questions the author had throughout the project. Co-authors affiliated with the Aerosol Physics group and the Finnish Meteorological Institute (FMI) attended meetings during which the project was discussed. FMI provided the data required for the experiments. The author developed the system, performed the experiments, evaluated the results, and wrote the manuscript. Co-authors, in particular Elham Baranizadeh, Harri Portin, Mika Komppula, Mauno Rönkkö, and Mikko Kolehmainen provided comments to the manuscript.

Paper **IV** was conceived following a presentation by Hanna Huitu on disease pressure modelling. The author suggested to use ideas developed in papers **II** and **III** and build a more advanced version of the system Hanna had presented. Specifically, the author suggested that the system could forecast disease pressure, model pest outbreaks as situations, and provide explicitly represented information about situations as well as a more interactive experience for system users, e.g. agricultural advisers and farmers, with situational knowledge located in space-time. The Natural Resources Institute Finland (LUKE), formerly MTT Agrifood Research Finland, provided the data and a textual description of the disease pressure model. The author implemented the model in software. Jussi Nikander acted as the primary collaborator on the interface between the two institutions and thus addressed most domain specific concerns through-

out the project. Markku Koistinen is acknowledged for suggesting to use FMI Open Data as the source for daily weather forecast data. The author developed the system, performed the experiments, evaluated the results, and wrote the manuscript. Co-authors attended meetings and provided comments to the manuscript.

Paper **V** resulted in work that aimed at grounding the architecture of the software framework proposed in the dissertation in an existing reference model. The selected reference model is an abstract system architecture for the 'archetypical' environmental research infrastructure. It is designed to model the data life-cycle in such infrastructure, including the acquisition of data, in particular from sensor networks, and the curation, access, and processing of data. Paper **V** extends the reference model with functionality for knowledge acquisition, and the curation, access, and processing of knowledge. The author developed the idea and wrote the manuscript. The co-authors Mauno Rönkkö and Mikko Kolehmainen provided comments to the manuscript.

Presentations

Parts of the dissertation were presented in the following international conferences and workshops:

- First Joint International Workshop on
  Semantic Sensor Networks and Terra Cognita
  Bethlehem, Pennsylvania, USA, October 11, 2015
- International Conference on
  Knowledge Engineering and Semantic Web
  Moscow, Russia, September 30 - October 2, 2015
- International Symposium on
  Environmental Software Systems
  Melbourne, Australia, March 25-27, 2015
- International Congress on
  Environmental Modelling and Software
  San Diego, California, USA, June 15-19, 2014
- International Symposium on
  Environmental Software Systems
  Neusiedl am See, Austria, October 9-11, 2013
- Third Workshop on
  Intelligent Systems for Quality of Life Information Services
  Halkidiki, Greece, September 29, 2012
- International Congress on
  Environmental Modelling and Software
  Leipzig, Germany, July 1-5, 2012
- International Symposium on
  Environmental Software Systems
  Brno, Czech Republic, June 27-29, 2011

Contents

Figures

Abbreviations

+K	Plus Knowledge
ABox	Assertional Box
AmI	Ambient Intelligence
ANN	Artificial Neural Network
API	Application Programming Interface
ASP	Answer Set Programming
CEP	Complex Event Processing
DMPS	Differential Mobility Particle Sizer
ENVRI	Common Operations of Environmental Research Infrastructures
ENVRI-RM	ENVRI Reference Model
EPL	Event Processing Language
ESFRI	European Strategy Forum on Research Infrastructures
EU	European Union
FMI	Finnish Meteorological Institute
FP7	Seventh Framework Programme
FSTO	Fuzzy Situation Theory Ontology
GIS	Geographic Information System
GLEON	Global Lake Ecological Observatory Network
GOOS	Global Ocean Observing System
GPS	Global Positioning System
HMM	Hidden Markov Model
ICOS	Integrated Carbon Observation System
IoT	Internet of Things
IRI	Internationalized Resource Identifier
JSON	JavaScript Object Notation
LTER	Long Term Ecological Research
LUKE	Natural Resources Institute Finland
MLP	Multilayer Perceptron (artificial neural network)
MMEA	Monitoring and Environmental Efficiency Assessment
MTT	MTT Agrifood Research Finland
NEON	National Ecological Observatory Network
O&M	Observations and Measurements
OGC	Open Geospatial Consortium
OWL	Web Ontology Language
PCT	Parsimonious Covering Theory
PROV	Provenance

Continued on next page

Continued from previous page

PROV-O	PROV Ontology
PWS	Present Weather Sensor
QB	RDF Data Cube Vocabulary
RDF	Resource Description Framework
RDFS	RDF Schema
REST	Representational State Transfer
SAX	Symbolic Aggregate Approximation
SensorML	Sensor Model Language
SMEAR	Station for Measuring Ecosystem-Atmosphere Relations
SOS	Sensor Observation Service
SPARQL	SPARQL Protocol and RDF Query Language
SQL	Structured Query Language
SSN	Semantic Sensor Network (ontology)
STO	Situation Theory Ontology
SVD	Singular Value Decomposition
SWRL	Semantic Web Rule Language
SYKE	Finnish Environment Institute
TBox	Terminological Box
URI	Uniform Resource Identifier
W3C	World Wide Web Consortium
WaterML	Water Markup Language
WKT	Well-Known Text
XML	Extensible Markup Language
XSD	XML Schema Definition

Hyperlinks

52°North	52north.org
Air Quality Egg	airqualityegg.com
Apache Cassandra	cassandra.apache.org
Apache Commons Math	commons.apache.org
Apache Storm	storm.apache.org
Apache Tomcat	tomcat.apache.org
ArcGIS	esri.com/software/arcgis
Complexible Inc	complexible.com
Conservation International	conservation.org
DBpedia	wiki.dbpedia.org
Emrooz	github.com/markusstocker/emrooz
ENVRI	envri.eu
ENVRI-RM	envri.eu/rm
Esper	espertech.com/esper
Excel	products.office.com/en-us/excel
FMI	en.ilmatieteenlaitos.fi
GeoServer	geoserver.org
GeoSPARQL	opengeospatial.org/standards/geosparql
GLEON	gleon.org
Google Directions API	developers.google.com
Google Earth	google.com/earth
Google Geocoding API	developers.google.com
GOOS	ioc-goos.org
ICOS	icos-ri.eu
JScience	jscience.org
JSON	json.org
LTER	lternet.edu
LUKE	luke.fi/en
Matlab	mathworks.com/products/matlab
MMEA	mmea.fi
NEON	neoninc.org
NetCDF	www.unidata.ucar.edu/software/netcdf
O&M	opengeospatial.org/standards/om
OGC	opengeospatial.org
OWL	w3.org/2001/sw/wiki/OWL

Continued on next page

Continued from previous page

OWL-Time	w3.org/TR/owl-time
Pellet	clarkparsia.com/pellet
PostgreSQL	postgresql.org
Profium Sense	profium.com/en
PROV-O	w3.org/TR/prov-o
QB	w3.org/TR/vocab-data-cube
RDF	w3.org/RDF
RDFS	w3.org/TR/rdf-schema
SensorML	opengeospatial.org/standards/sensorml
SMEAR	www.atm.helsinki.fi/SMEAR
SOS	opengeospatial.org/standards/sos
SPARQL	w3.org/TR/rdf-sparql-query
SSN	w3.org/2005/Incubator/ssn/ssnx/ssn
Stardog	stardog.com
STO	vistology.com/ont/2008/STO/STO.owl
SWRL	w3.org/Submission/SWRL
SYKE	syke.fi/en-US
Tekes	tekes.fi/en
W3C	w3.org
WaterML	opengeospatial.org/standards/waterml
Wavellite	uef.fi/en/envi/projects/wavellite
WEKA	cs.waikato.ac.nz/ml/weka
XML	w3.org/XML
XSD	w3.org/TR/xmlschema-0

Accessed on October 18, 2015.

Errata

1. The disjointness axiom for personal car and truck classes in Figure 4.1 is wrong and should be

$$\text{PersonalCar} \sqcap \text{Truck} \sqsubseteq \bot$$

2. In Paper **I**, Figure 3, the band-pass filter frequency interval 80-130 Hz is wrong and should be 100-160 Hz. The references to Figure 5(b) in text at pp. 4 and 8 (cols. 2) are wrong and should be to Figure 6(b). The note in Table 5 wrongly refers to Figure 6(b) instead of Table 6(b). The discussion section (para. 1) states that we "compared [our results] with the results of similar studies published in the literature" but this comparison is discussed only later in the section. Finally, the publisher introduced inconsistencies in writing style, e.g. 600 s vs. 8.192s, data set vs. dataset, or event-of-interest vs. event of interest.

3. In Paper **II** (p. 1442, col. 2) the Resource Description Framework is stated to be a knowledge representation language. This is inaccurate and the corresponding sentence is rephrased as: "Situational knowledge acquired from vibration data was represented in a domain ontology, using the Web Ontology Language (OWL) [23] knowledge representation language and the Resource Description Framework (RDF) [24] data model."

4. In Paper **II** (p. 1443, col. 2) and Paper **III** (p. 31, col. 2) the index i of the first object a_i in infon $\ll R, a_i, \ldots, a_m, i \gg$ is wrong and should be 1. According to Devlin (1991, p. 115) the infon is

$$\ll R, a_1, \ldots, a_m, i \gg$$

5. Paper **III** (p. 28, col. 1; p. 34, col. 2; and p. 37, col. 1) cites the manuscript Stocker et al. "The [W]avellite modelling and software framework for situation awareness in environmental monitoring" submitted to Environmental Monitoring and Assessment from which it was eventually withdrawn. The manuscript was later submitted to other journals which, however, rejected the submission, primarily on the grounds that the framework was already published in Paper **III**. The manuscript was never published. Instead, we include some of the content of the manuscript, specifically the description of the Wavellite foundations, in this dissertation (primarily Section 3.3.2, "Information Viewpoint"). Paper **III** (p. 29, col. 2) refers to sensing devices as

procedures. This was inspired by the OGC O&M standard which lists sensor as a type of procedure. In retrospect, it would have been better to merely talk about SSN sensors and sensing devices. The publisher introduced the following errors. At p. 31, col. 2, a blank space is missing between i_m and 'is' in "[...] argument places $i_1, \ldots, i_m$is sufficient [...]" (corr.: $i_1, \ldots, i_m$ is); at p. 34, col. 1, the parameter $\dot{t_1}$ is missing in "[...] cloud event begins and ends (and $\dot{t_2}$) [...]" (corr.: ($\dot{t_1}$ and $\dot{t_2}$)); and at p. 34, col. 1, a period is missing after "[...] and mean precipitation $6.4\,\mathrm{mm\,h^{-1}}$" (corr.: $6.4\,\mathrm{mm\,h^{-1}}$.) Finally, at p. 33, col. 1, the motivation is three-fold (not three-folded).

6. Paper **V** (p. 14, col. 2; p. 15, col. 1) cites the manuscript Stocker et al. "Plant disease pressure situation modelling in agriculture" as being in review at Computers and Electronics in Agriculture. Unable to obtain more than one formal review, the journal later rejected the submission. The manuscript (i.e. Paper **IV**) is currently submitted to Environmental Modelling & Software. At p. 6, col. 1, there is an excessive 'a' in "[...] such as a the fact that [...]" (corr.: such as the fact that).

1 Introduction

In 2005, Gartner predicted that "[b]y 2015, wirelessly networked sensors in everything we own will form a new [w]eb" and specified that the new web "will only be of value if the 'terabyte torrent' of data it generates can be collected, analyzed and interpreted" (Raskino et al., 2005).

It is 2015 and wirelessly networked sensors are arguably not in *everything* we own. Compared to 2005, sensors are, however, found in more of what we own. Smart phones are perhaps the prime example but wirelessly networked sensors are today part of many consumer products, including the Air Quality Egg and the growing assortment of wearable technology used, e.g., in sports (Baca et al., 2009) or health monitoring (Pantelopoulos and Bourbakis, 2010).

The conditional in Raskino et al.'s prediction is particularly interesting. Obviously, having sensors monitoring us and our environment is of no value if their data cannot be collected, analysed and interpreted. The observation is arguably as valid and urgent in 2015 as it was in 2005. While network, storage, and database technologies have advanced to address the collection and management of the 'terabyte torrent', the (automated) analysis and interpretation of the collected data have been, and continue to be, an open challenge (Kimani et al., 2004; Compieta et al., 2007). The "complexity of the function relating data to description" (Herbin et al., 2012) is a critical problem beyond data collection and management, and is a hint for why the analysis and interpretation of the 'terabyte torrent' largely remains an open issue.

The challenge persists also in systems for environmental monitoring. In this dissertation, environmental monitoring systems include hardware, software, and people, and their purpose is to provide us with data, collected using environmental sensor networks (Martinez et al., 2004). The data are processed using statistical analysis and modelling to study and hopefully advance our understanding of the environment (Balazinska et al., 2007; Devaraju et al., 2014). Understanding is desirable toward various ends, including research, policy, and education.

Environmental monitoring systems continue to be designed and deployed to monitor an increasingly large number and heterogeneous type of environmental phenomena. To draw a few examples, sensor networks are used to monitor the atmosphere for gas and particle concentration, often at high temporal resolution, with resulting data serving, e.g., research and air quality information systems. Atmospheric phenomena such as new particle formation are studied for their impact on the climate and human health. The weather on agricultural land is monitored to anticipate pest outbreaks and to support farmer decision making.

Environmental monitoring systems also continue to increase in complexity and spatial coverage, with modern systems, e.g. the U.S. National Ecological Observatory Network, spanning continental-scale environmental sensor networks and other systems reaching global scale, e.g. the Global Ocean Observing System or the Global Lake Ecological Observatory Network.

According to Cregan (2007), "[i]t is generally agreed that machines should be doing more of the work of turning data into knowledge in a way that supports the production of results for human benefit." Turning data into knowledge has been termed 'closing the semantic gap' and 'sense making', and the problem was recognized in various domains, including robotics, context awareness, and ambient intelligence. System architectures have been proposed to address the problem. Methods in digital signal processing, machine learning, knowledge representation, machine vision, and machine reasoning have been adopted in architectures and system implementations.

Inspired by the idea of machines "doing more of the work of turning data into knowledge," we develop, implement, and validate on case studies a software process for automated and near real-time acquisition of situational knowledge from data, and the representation and processing of situational knowledge, in environmental monitoring systems.

Data are for properties of environmental phenomena and are generally collected from environmental sensor networks that monitor properties over time and space. For instance, an environmental sensor network may monitor the property of concentration of environmental phenomena such as particulate matter ($PM_{2.5}$ and PM_{10}) and gases (O_3 and NO_2) in a volume of ambient air and over time.

Knowledge is about situations with phenomena as their objects, is obtained from data, and is represented using knowledge representation languages and technologies. Situations are structured parts of reality (Devlin, 1991). Structure is in the relations among relevant objects of the part of reality. Situations are of the monitored environment and are observed by environmental monitoring systems. For instance, an episode of unhealthy exposure to ambient air is a situation. Relevant objects include a population, particulate matter, gases, space, and time. A population is exposed to particulate matter and gases in particular volumes of space-time. Episodes of unhealthy exposure are observed by environmental monitoring systems.

Computational methods in digital signal processing, data-driven modelling, and physically-based modelling are utilized to process data and acquire situational knowledge from data. The proposed software process is implemented in a software framework. The framework supports the development of systems for situation assessment capable of obtaining and maintaining situation awareness in environmental monitoring. An architecture for the software framework is presented. Building on the framework, we develop situation-aware environmental monitoring systems for three case studies.

The case studies are in intelligent transportation systems, atmospheric science, and agricultural science. They discuss non-trivial situational knowledge acquisition problems in environmental monitoring, on data acquired from environmental sensor networks or data resulting from models.

The remaining sections in this chapter present the research question, research objectives, and contributions. Chapter 2 introduces the relevant concepts, namely environmental monitoring, situation awareness, situation theory, ontology, and modelling. Chapter 3 presents the architecture and implementation of the proposed software framework. Chapter 4 provides an integrated overview of the applications we developed using the software framework. The applications are presented in details in papers **II**, **III**, and **IV**. Chapter 5 discusses related work, the strengths and limitations of the proposed approach, and future work. Chapter 6 concludes with a summary and final remarks.

1.1 RESEARCH QUESTION

The dissertation explores the following research question. The question takes environmental monitoring systems as the unit of analysis and suggests that it may be possible for such systems to use a particular theory to model situations, and a particular set of technologies to acquire and represent knowledge about situations. The research question is phrased as follows:

> *Can environmental monitoring systems utilize situation theory to model observed situations, and utilize ontology and related technologies to represent situational knowledge obtained from data processed by means of computational models?*

The research question is reworded as the following two claims, labelled **C1** and **C2**. We aim at validating these claims by developing and evaluating environmental monitoring systems for the selected case studies.

> *Environmental monitoring systems can utilize situation theory to model observed situations.* **(C1)**

Claim **C1** makes two assumptions. First, it is assumed that there exist situations, i.e. structured parts of reality. Second, it is assumed that such situations are of a monitored environment and can be observed by an environmental monitoring system, i.e. a physical-socio-technical system consisting of a physical environment and of human, hardware, and software agents. The claim then proposes that the system can utilize situation theory (Barwise and Perry, 1981; Devlin, 1991) to model situations.

> *Environmental monitoring systems can utilize ontology and related technologies to represent situational knowledge obtained from data processed by means of computational models.* **(C2)**

Claim **C2** makes three assumptions. First, it is assumed that knowledge about situations can be obtained. This problem is fundamental to situation theory. Second, it is assumed that it is possible to obtain situational knowledge from data. Third, it is assumed that data can be processed by means of computational models. A particular class of models are those that process data to extract knowledge. The claim then proposes that the system can utilize ontology, as understood in information science, and related technologies in knowledge representation and reasoning to represent situational knowledge.

1.2 RESEARCH OBJECTIVES

Situation awareness is commonly understood as the perception, comprehension, and projection of what is going on around us (Endsley, 1995). Over the past two decades, the concept has found application in domains and problems where human operators require awareness about the state of elements in certain space-time volumes. Classical examples are applications in aviation, military, driver assistance, marine port surveillance, airport control, emergency and rescue management. The application of situation awareness to environmental monitoring, in particular monitoring in environmental science, is relatively uncommon—even though citizens and researchers who consume data and information obtained in environmental monitoring arguably do so because they are interested in knowing what is going on around them, e.g. obtain air quality information to decide whether or not to go jogging.

A first objective of this dissertation is to interpret the state of understanding situations observed by environmental monitoring systems as situation awareness, interpret the process of achieving and maintaining such understanding as situation assessment, and leverage on existing situation awareness models in the design of an architecture for a software framework that supports the development of systems for situation awareness in environmental monitoring. This objective is reflected in the following claim, labelled **C3**.

> *Environmental monitoring systems can utilize environmental sensor networks to perceive the properties of phenomena in situations, and utilize computational methods to comprehend and project situations, and involved phenomena. Such systems implement the process of situation assessment to obtain and maintain awareness about situations of a physical environment.* **(C3)**

In situation theory, situations are structured parts of reality. Parts of reality consist of objects. Objects stand in relations. Relations build structure. A second objective is to model that which is observed by environmental monitoring systems as structured parts of reality, i.e. situations. Objects are, generally, environmental phenomena, including spatial and temporal locations.

Situation theory models situation as a mathematical structured object. Situation theory thus provides a formalization of the concept of situation and information about situations. A third objective is to utilize this mathematical object as a model for situational knowledge objects obtained and maintained in environmental monitoring systems.

Situational knowledge objects can be represented using ontology, and technologies such as ontology languages. Of particular interest is the situation theory ontology (Kokar et al., 2009). This ontology provides a machine readable and interpretable formalization of the concept of situation, suitable for representing and processing situational knowledge objects in environmental monitoring systems. A forth objective is to design an ontological framework, centred around the situation theory ontology, for the representation of knowledge about situations, including temporal and spatial locations. The ontological framework serves in the representation of situational knowledge acquired and maintained by environmental monitoring systems.

Situational knowledge is acquired from data processed by means of computational models. A fifth objective is to design an architecture for a software framework that supports the acquisition of situational knowledge from data and the representation of situational knowledge. The architecture is expected to build on the ontological framework for the representation of situational knowledge, and to extend this framework with further ontologies required by the architecture to represent relevant raw and processed data objects. Specifically, environmental monitoring systems need to acquire raw sensor data from environmental sensor networks, represent sensor data, and curate such data using suitable data management systems. Systems are also required to support data access interfaces, data processing, and the representation and curation of processed data. A sixth objective is to propose an open source implementation for the software framework architecture.

The acquisition, curation, access, and processing of data is functionality that an environmental monitoring system has at least partially in common with environmental research infrastructure. A seventh objective is to extend the ENVRI reference model (Chen et al., 2013a) for the 'archetypical' environmental research infrastructure with functionality for knowledge acquisition from curated data, and the curation, access, and processing of knowledge. An eight objective is to ground the architecture of the proposed software framework in the extended ENVRI reference model. This objective aligns the proposed software framework with environmental research infrastructure. The alignment is argued to support the following additional claim, labelled **C4**.

> *Environmental research infrastructure can support the acquisition, curation, access, and processing of (situational) knowledge. If an environmental research infrastructure supports such functionality it is a knowledge-based environmental research infrastructure.* **(C4)**

Finally, a ninth objective is to utilize the software framework implementation to develop situation-aware environmental monitoring system applications. The applications address situational knowledge acquisition problems in intelligent transportation systems, atmospheric science, and agricultural science. Their purpose is to validate the claims **C1** and **C2**, and to support the claims **C3** and **C4**. Each application is an environmental monitoring system consisting of an observed physical environment, sensor networks and other hardware agents, program logic and models and other software agents, and human agents, including software engineers and scientists. Each application utilizes situation theory to model relevant situations of the physical environment observed by the system, and utilizes the ontological framework to represent situational knowledge obtained from data processed by means of computational models. Applications may represent raw sensor data and processed data. The applications also demonstrate situational knowledge processing, such as situational knowledge visualization or situation projection.

1.3 CONTRIBUTIONS

State of the art environmental research infrastructures have primarily addressed data life-cycle management, from data acquisition to data processing. The ENVRI reference model suggests that such infrastructures do not handle information and knowledge obtained from processed data.

We suggest that environmental research infrastructure can address knowledge life-cycle management, in particular the acquisition, curation, access, and processing of situational knowledge. A contribution of this dissertation is the proposal to extend the ENVRI reference model with a model for the knowledge life-cycle in environmental research infrastructure. The result is knowledge-based environmental research infrastructure. To the best of our knowledge, the proposed extension to the ENVRI reference model and the resulting notion of knowledge-based environmental research infrastructure are novel contributions.

A second contribution is the architecture of a software framework designed for the acquisition, curation, access, and processing of environmental monitoring data and situational knowledge. An open source implementation for the software framework architecture is a third contribution. To the best of our knowledge, the proposed approach is novel for its support of automated situational knowledge acquisition in environmental monitoring, particularly in scientific applications, using methods in digital signal processing, machine learning, and knowledge representation and reasoning.

The proposed approach is evaluated and discussed for three case studies with applications for situational knowledge acquisition and processing about (1) road vehicles observed in data by a road-pavement vibration sensor network; (2) atmospheric new particle formation observed in data by a differential mobil-

ity particle sizer as well as cloud events observed in data by a present weather sensor; and (3) disease outbreaks in agricultural crops computed using a disease pressure model and weather forecast data. The development of these case studies with non-trivial situational knowledge acquisition problems is a fourth contribution. The application of the proposed approach to intelligent transportation systems, atmospheric science, and agricultural science is arguably novel.

2 Concepts

We present the central concepts underlying the dissertation. Section 2.1 introduces environmental monitoring. Environmental monitoring is the domain addressed by the research question and for which we design, implement, and evaluate software systems. The section frames the problem and motivates the dissertation. Section 2.2 introduces situation awareness. Situation awareness is the state which environmental monitoring systems are claimed to be capable of obtaining and maintaining. The section draws on different models of situation awareness. We suggest that the concept of situation awareness is useful in environmental monitoring. We also suggest that various elements of situation awareness models can guide the design of system architectures for data acquisition and processing, as well as situational knowledge acquisition, representation, and processing in environmental monitoring. Section 2.3 introduces situation theory. Situation theory proposes a mathematical ontology for knowledge (information) about situations. The theory is relevant for the modelling of the structured parts of reality observed by environmental monitoring systems. Section 2.4 introduces ontology, as understood in information science. Ontologies, ontology languages, and related technologies are relevant because they support the representation and processing of situational knowledge in environmental monitoring systems. Finally, Section 2.5 introduces modelling and models, in particular computational models. Models, specifically empirical and physically-based models, are relevant primarily because they enable the acquisition (or extraction) of situational knowledge from data processed in environmental monitoring systems.

2.1 ENVIRONMENTAL MONITORING

Environmental monitoring is the domain addressed by the research question and for which we design, implement, and evaluate software systems. Meijers (1986) defines monitoring as:

Definition 2.1.1 (Monitoring)**.** The process of repetitive observing, for defined purposes on one or more elements of the environment according to prearranged schedules in space and time and using comparable methodologies for environmental sensing and data collection.

Discussing modifiers used in connection with monitoring, Meijers specifies that "environmental and ecological monitoring are concerned with the natural environment." Specifically, environmental monitoring focuses on physical and chemical entities while ecological monitoring focuses on ecological entities.

According to Definition 2.1.1, monitoring is the process of repetitive observing. The definition does not further specify forms of observation. Observation may for instance be in form of a parent watching a child building a castle in a sandbox; a scientist performing chemical analyses to estimate the concentration of chlorophenols in soil samples; a thermometer measuring the temperature of ambient air. In this last example, observation is made by an instrument. Instrumental observation is of specific interest in this work, and the instruments are called sensing devices.

The example for a thermometer monitoring the temperature of ambient air introduces the concept of measurement. Finkelstein (1982, p. 6) defines measurement as:

Definition 2.1.2 (Measurement)**.** The process of empirical, objective, assignment of numbers to properties of objects or events of the real world in such a way as to describe them.

Measurement, as defined by Finkelstein, constrains the form of observation to the assignment of numbers to properties. The properties are of elements of the environment and, in environmental monitoring, the properties are of physical or chemical entities. Unless stated otherwise, it is assumed here that the process of measurement is implemented by sensing devices; that a sensing device is embedded in the environment (i.e. *in situ*), and measures a property (e.g. temperature) of a phenomenon, also known as feature of interest (e.g. ambient air); and that the numbers assigned to properties in the process of measurement are digital and are generally referred to as sensor data or sensor observation values.

Following the design pattern introduced by Janowicz and Compton (2010), entities that relate sensing devices, sensor data, measured properties of features, time and space are called observations. Janowicz and Compton classify observations as social, rather than physical, objects, i.e. as objects with setting in communication events. A formalization of the concept is discussed in Chapter 2.4. The notion of measurement is thus different from observation. Most importantly, measurement is a process while observation is an object. Moreover, observations are not merely the numbers assigned to properties in the process of measurement. Rather, observations are objects with a complex relational structure.

2.1.1 Sensor Networks

Following Definition 2.1.1, monitoring is for one or more elements of the environment and extends in space and time. A single sensing device typically monitors (a specific property of) a particular element in time. To monitor several properties of various elements in both space and time, environmental sensor networks can be embedded in the environment.

Environmental sensor networks are specialized sensor networks (Akyildiz et al., 2002; Chong and Kumar, 2003) tailored for environmental applications (Hart and Martinez, 2006). Hart and Martinez provide a brief account of the evolution of environmental sensor networks from passive logging systems that require manual data downloading to active sensor networks that communicate data automatically, increasingly using wireless technology.

Hart and Martinez (2006) discuss a generic architecture for environmental sensor networks. The architecture consists of an array of sensor nodes (i.e. devices containing sensors) that gather data autonomously, and a communication network through which data flows via one or more base stations to a server, called sensor network server. In this generic architecture, the role of base stations is to support a group of sensor nodes and to forward the group's data. The sensor network server is the sink for data and its role is to make the data available, on the Internet and for visualization, integration, and analysis (Martinez et al., 2004; Hart and Martinez, 2006). The authors underscore that, as we move from sensor nodes to the sensor network server, the mobility of the components decreases whereas the computational power, data storage, and power availability increase.

Having the potential to advance our understanding of the natural world, including the possibility of revealing so far unobserved phenomena, environmental sensor networks have been argued to be invaluable research tools for earth and environmental science (Porter et al., 2005; Hart and Martinez, 2006; Collins et al., 2006; Lovett et al., 2007; Rundel et al., 2009; Benson et al., 2010). Examples for environmental sensor networks, national research programs that maintain such networks, and international organizations that form 'networks of networks' are abundant. Hart and Martinez review several dozen environmental sensor networks, which they classify in three categories—large scale single function, localised multifunction, and biosensor networks—plus the category of heterogeneous sensor networks, which aim to integrate data of various network types. Some well-known examples include the US Long Term Ecological Research Program (Michener et al., 2011, LTER), the US National Ecological Observatory Network (Keller et al., 2008, NEON), and the Global Lake Ecological Observatory Network (Kratz et al., 2006, GLEON).

Examples for environmental monitoring programs, including programs that utilize environmental sensor networks, are abundant also in Finland. To name a few, the Finnish Environment Institute (SYKE) maintains monitoring programs related to the Baltic Sea, inland waters, and water resource management as well as programs related to ecosystem services and biodiversity. The Finnish Meteorological Institute (FMI) operates over 400 observation stations around Finland to monitor, among other phenomena, weather, ambient air, and radioactivity. The Natural Resources Institute Finland (LUKE) operates networked weather stations at selected farms for research in precision agriculture (Thessler et al., 2011).

Since 2005, the Puijo tower in Kuopio, Finland, serves as platform for a semi-urban measurement station to observe "several meteorological parameters, aerosol and cloud droplet size distribution, aerosol optical properties and trace gas concentrations" (Leskinen et al., 2009). The data obtained in monitoring serves research in cloud formation, aerosol-cloud interaction, and atmospheric aerosol particle formation. The measurement station at Puijo is part of the Station for Measuring Ecosystem-Atmosphere Relations (SMEAR) research program. SMEAR infrastructure implements "continuous, comprehensive measurements of fluxes, storages and concentrations in the land ecosystem-atmosphere continuum" (Hari and Kulmala, 2005). SMEAR aims at understanding the coupling and feedback between the atmosphere, vegetation and soil in order to "provide more accurate projections of future atmospheric composition" (Hari and Kulmala, 2005).

Within the Measurement, Monitoring and Environmental Efficiency Assessment (MMEA) research program, Finland has recently also invested considerable resources to develop new sensing technologies, a cloud-based software platform for data integration and processing, and applications and services that utilise the platform. Challenges faced by the MMEA program include the design and implementation of a software platform that supports the connection and integration of observation and model (streamed) data from potentially hundreds of sources, ranging from sensing devices to web services; the processing of such data within the platform to support, e.g., quality control or complex event detection; and the (streamed) distribution of integrated processed data to client applications.

Environmental monitoring can be a source of large amounts of heterogeneous data. This is self-evident for monitoring that involves large-scale networks with heterogeneous sensing devices. According to Michener et al. (2011), within the US LTER Network data volumes in the range of 10s to 100s gigabytes are acquired manually and automatically on a weekly to annual basis. Battams (2014) reports that the Solar Dynamics Observatory "returns in excess of 1 terabyte of data daily." The SMEAR II station in Hyytiälä, Finland, generates approximately 1 gigabyte of data per day (personal communication). Monitoring using localised sensor networks can lead to large amounts of, possibly heterogeneous, data, too. In monitoring the micro climate in the canopy of a single coastal redwood over 44 days using a wireless sensor network consisting of 33 devices deployed into the tree, Tolle et al. (2005) underscore that the upper bound data yield is 1.7 million data points. Benson et al. (2010) note that a single instrumented lake buoy can generate megabytes of data per day, and a flux tower exceeds these volumes. The heterogeneity of data is largely a consequence of operating diverse sensing device types and systems, e.g. temperature and humidity sensors, automated weather stations, eddy-covariance flux towers, radar. Data of such systems are encoded, formatted, and accessed in heterogeneous ways. Combined manual and automatic measurement also increases heterogeneity.

2.1.2 Challenges

The heterogeneous, typically voluminous and possibly real-time streamed, data in environmental monitoring continue to pose challenges to data organization and interpretation (Collins et al., 2006; Rundel et al., 2009; Nittel, 2009; Benson et al., 2010; Michener and Jones, 2012). These challenges point to open issues in environmental monitoring, in particular monitoring based on environmental sensor networks.

In some environmental monitoring programs, such as LTER and NEON, data acquisition can occur via on-site manual sampling, e.g. monitoring plant development or collecting insects; in laboratories, e.g. monitoring chemical concentrations in soils; and automatically via environmental sensor networks, e.g. monitoring atmospheric properties (Michener et al., 2011). In manual data acquisition, paper data sheets and tape recorders are common and are examples for techniques that require subsequent data entry (Michener et al., 2011). In contrast, with environmental sensor networks data acquisition can be automated. Different techniques require different approaches to data acquisition. For instance, for large data volumes automating quality assurance/quality control is essential (Rundel et al., 2009). The acquired data are thus not just voluminous and heterogeneous but data acquisition may rely on fundamentally different processes.

Although widely utilized to organize environmental monitoring data, traditional relational database management systems have important shortcomings in environmental monitoring (Carney et al., 2002; Madden and Franklin, 2002). Sensor network characteristics, such as limited power and computational resources, have motivated novel strategies for data acquisition (Madden et al., 2003). Furthermore, environmental monitoring data are rarely updated (Michener et al., 2011) and continuous data streams require operators not supported by traditional relational database management systems, such as windowed operators (Carney et al., 2002). Resolving the syntactic and semantic heterogeneity of environmental monitoring data is a further challenge in data organization (Horsburgh et al., 2009).

Beyond organization, the interpretation of environmental monitoring data is another important challenge. The value of data is arguably in the knowledge about observed environmental phenomena that can be obtained from data. Thus, the organization of environmental monitoring data is merely an intermediate step that serves data interpretation. The result of data interpretation is information, from which knowledge is learned (Aamodt and Nygård, 1995). Data interpretation poses a range of additional questions. Is it performed manually or can it be done automatically? What are suitable methods? How involved is the process? What information do we obtain from data interpretation, and how is information and knowledge represented?

When aerosol scientists use data visualization techniques to detect at which days atmospheric new particle formation occurs, classify the events according to a classification scheme, and characterize the events for their attributes such as event start and end times (Hamed et al., 2007), unless represented explicitly the obtained knowledge remains implicit in the data. Aerosol scientists present their results and conclusions in articles, in natural language text using tables and figures. In such forms, information and knowledge is unfortunately hardly accessible to computer systems. Furthermore, the knowledge obtained from data is generally summarized—condensed into, and presented as, summary statistics. Claim **C2** suggests that computer systems can represent knowledge about situations and involved environmental phenomena, such as atmospheric new particle formation, obtained from monitoring data explicitly and formally, using ontologies, ontology languages, and related technologies (introduced in Section 2.4). Furthermore, computer systems can automate the extraction and the explicit, formal, representation of knowledge—and, therefore, automate the interpretation of environmental monitoring data.

Knowledge about the monitored environment that can be obtained from environmental sensor network data is diverse. It is thus useful to identify a unifying abstraction for such knowledge. Claim **C1** suggests that the concept of *situation* is a suitable candidate for such a unifying abstraction in environmental monitoring. In this dissertation, knowledge is, therefore, generally situational knowledge, and monitored environmental phenomena are objects in situations.

Section 2.3 introduces situation theory and its formal definition of situation. The following section first introduces situation awareness and, in particular, two models of situation awareness that have been proposed in the literature. These models can inspire the design of system architectures for data acquisition and processing as well as situational knowledge acquisition, representation, and processing in environmental monitoring.

2.2 SITUATION AWARENESS

Situation awareness has been said to be "knowing what is going on around you" (Endsley, 2000). Knowing what is going on is arguably often the motivation at the base of operating an environmental sensor network. Such networks support knowing what is going on in observed environments. Endsley (1995) defines situation awareness as:

Definition 2.2.1 (Situation Awareness)**.** The perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future.

Situation awareness is a state of knowledge, and Endsley explicitly distinguishes situation awareness from situation assessment, i.e. the state from the processes used to achieve, acquire, or maintain the state.

Definition 2.2.1 consists of three levels, underscored by the three keywords perception, comprehension, and projection. Accordingly, Endsley's model is called the three level model. The first level is concerned with perceiving the "status, attributes, and dynamics of relevant elements in the environment" (Endsley, 1995). Of interest to perception are the relevant characteristics of elements located in a volume bounded by time and space. The second level is concerned with understanding the "significance of those elements in light of pertinent operator goals" (Endsley, 1995). At the first level, perception of relevant elements results into important information (Endsley, 2000). At the second level, information is integrated and interpreted in order to determine their relevance to operator goals (Endsley, 2000). Operator and operator goals are key components of the model in that the operator is the individual, holder of situation awareness, and awareness is "fundamentally linked with [an individual's] goals" (Endsley, 1995). It is operator goals that specify which elements in the environment, and which element characteristics, are relevant. The third level is concerned with projecting "the future actions of the elements in the environment" (Endsley, 1995). Projection supports the operator in deciding how to respond to future actions of the elements such that goals can be met.

In Endsley's model, the individual, i.e. the human operator or expert, is the unit of analysis (Stanton et al., 2010). The model does provide for the possibility of technical systems to mediate between the environment and the individual, e.g. as implementers of perception. However, holders of situation awareness are individuals, and Endsley has been critical of automation in situation awareness (Endsley, 1996).

The three level model has been popular over decades. However, there exist alternative models of situation awareness that address what have been argued to be important limitations of the three level model. One such alternative, presented next, is of particular interest here.

2.2.1 Alternative Model

Discussing three viewpoints of situation awareness—the psychological, the engineering, and the systems ergonomics—Stanton et al. (2010) argue that situation awareness in Endsley's three level model is an individual psychological phenomenon of human information processing. Hence, the model follows the psychological approach to situation awareness. Stanton et al. and Salmon et al. (2007) have criticized the psychological approach and the three level model, in particular. Stanton et al. contend that the three level model inadequately explains "behaviour outside the unit of analysis." Specifically, the authors argue that the three level model is ill-suited for describing team behaviour, and situation awareness with teams as the unit of analysis, rather than individuals. The critique is valid also if socio-technical systems are the unit of analysis in situa-

tion awareness. Though Endsley agrees that there is a role for technical systems, the three level model seems to exclude the possibility for technical systems to be agents in situation awareness and part of the unit of analysis.

Stanton et al. (2006) propose a model for distributed situation awareness that follows the systems ergonomics approach to situation awareness, and has socio-technical systems as the unit of analysis. Such systems consist of both human and non-human agents. In their model, individual agents may have their own awareness of a situation, hold a particular view and associated knowledge, which can be different among different agents. It is the socio-technical system as a whole that holds all relevant knowledge, and has thus its own situation awareness.

The Stanton et al. model for distributed situation awareness with socio-technical systems as the unit of analysis is interesting for situation awareness in environmental monitoring. Environmental monitoring systems can arguably be understood as physical-socio-technical systems. System components include the monitored environment, human agents, and non-human agents. For instance, the coastal redwood, its canopy, and the surrounding micro climate studied by Tolle et al. (2005) is the monitored environment and the physical subsystem of the environmental monitoring system. The wireless sensor network with its 33 devices deployed into the tree are non-human agents and part of the technical subsystem. Computers and software are also non-human agents and thus parts of the technical subsystem. Finally, experts who conduct the study, analyse the data and publish the results are human agents and part of the social subsystem.

Agents communicate and have their own goals and views on situations. For instance, the goal of Vaisala HUMICAP© humidity and temperature probes operated by LUKE at farms in Finland is to measure two properties of ambient air at particular locations over time. Their view is on the signal obtained in measurement of the properties. They may convert the signal into digital data and communicate data over networks to software systems, e.g. a server with a database system. The goal of a database system is to manage data, and its view is on integrated data for the signals obtained in measurement of various properties, features, and locations. The database system communicates data to a human expert, e.g. in visualizations. The goal of a human expert is to interpret integrated data, and her view is on a particular phenomenon under study. She concludes that the weather during the growing season was too hot and dry, which negatively affected agricultural production. Another human expert may interpret the same data with view on a different phenomenon, e.g. crop pathogens, to conclude that pathogens respond differently to heat and drought. A human non-expert in agricultural science could interpret the data and conclude that the summer was ideal for sauna and swimming at a nearby lake.

2.2.2 Remarks

We can discuss this example scenario for ambient air monitoring to highlight some of the tenets of the distributed situation awareness model. First, both human and non-human agents hold some level of situation awareness, or at least contextually relevant information. Second, the different agents have different views on a volume of time and space, a structured part of reality. Third, with a shared goal, e.g. to study the effects of temperature and humidity on farming, the situation awareness of the two human experts can overlap. Having different goals, their situation awareness does however not overlap with the awareness held by the human non-expert interested in sauna and swimming. Fourth, the relationship between agents is held together by their level of situation awareness and goals, e.g. to study the effects of weather parameters on farming. Thus, situation awareness holds the loosely coupled environmental monitoring system together. The distributed situation awareness model predicts that, as a complex problem-solving system, an environmental monitoring system has its own situation awareness which cannot be accounted for by individual situation awareness (Stanton et al., 2006). With the distributed situation awareness model, Stanton et al. suggest a *systems* approach to situation awareness and clearly depart from the notion that situation awareness can only reside in the mind of a person. However, in their presentation of the model the authors inherit Endsley's three levels—perception, comprehension, and projection—with the difference that in distributed situation awareness non-human agents may participate in all three levels.

The hierarchical structure of the three level model is arguably a useful abstraction that can inspire the architecture of software systems for situation awareness in environmental monitoring. Furthermore, Endsley's distinction between situation assessment and situation awareness is also useful for such systems. This is perhaps unsurprising, given that the three level model is grounded in information processing theory (Salmon et al., 2007). Roughly speaking, a software system for situation awareness in environmental monitoring needs to acquire and process data obtained in monitoring to perceive relevant elements in the environment; it needs to explicitly represent and integrate information about relevant elements to comprehend their meaning in regard to goals; and it needs to manipulate integrated information to project the future state of relevant elements. Processes implemented by software systems to obtain and maintain situation awareness correspond to situation assessment. As a complex problem-solving system, environmental monitoring systems can be argued to hold their own situation awareness.

Human agents such as scientists are fundamental components of environmental monitoring systems. It is arguably human agents who, traditionally, achieve and maintain situation awareness in environmental monitoring. We ex-

plore how non-human agents of environmental monitoring systems, specifically software systems, can autonomously obtain and maintain situation awareness, and how situation awareness held individually can be shared among the social and technical components of an environmental monitoring system. To maintain and share situation awareness, it is necessary for software systems to implement situational knowledge representation.

2.3 SITUATION THEORY

This section introduces situation theory and its formal definition of situation. The concept is at the base of a knowledge object that serves as unifying abstraction for knowledge in situation-aware environmental monitoring systems.

Situation theory was developed by Jon Barwise and John Perry in the 1980s. Devlin (2004) recounts its development between 1980 and the early 1990s, as published by Jon Barwise, in part with John Perry. The theory has its foundations in situation semantics, a mathematically based theory of natural language semantics introduced by Jon Barwise in 1980 and developed by Barwise and John Perry. According to Devlin (2006), during the course of the 1980s, situation semantics became the "analysis of semantic issues of natural language based on situation theory," and situation theory served as the mathematical ontology of situation semantics. Devlin quotes Barwise and Perry (1980) who wrote:

> *The world consists not just of objects, or of objects, properties and relations, but of objects having properties and standing in relations to one another. And there are parts of the world, clearly recognized (although not precisely individuated) in common sense and human language. These parts of the world are called situations.*

In his book entitled *Logic and Information*, Devlin (1991) informally defines situation as:

Definition 2.3.1 (Situation, informal)**.** A situation is a structured part of reality that [an agent] somehow manages to [individuate].

Barwise and Perry's emphasis on the relations among world objects is arguably reflected in Devlin qualifying *parts* as being structured. Situations are "real, actual parts of the world" (Devlin, 2004, 2006). They are "basic and ubiquitous [and humans] are always in some situation or other" (Barwise and Perry, 1981). Situations are recognized, individuated, and Devlin (1991) attributes recognition to agents with "sophisticated cognitive abilities." As noted by Devlin (2006), the word *parts* is significant because a situation is limited to the objects at a location, i.e. a connected region of space-time (Barwise and Perry, 1981), and the information that an agent can obtain about a situation is limited. Devlin

(2006) highlights that situation semantics is an information based theory, as it studies the information exchanged when agents "in limited parts of the world [...] talk about [...] other limited parts of the world."

Example 2.3.1 (Situation)**.** The following scene is an example situation. A mother and her 5-year old boy are at a playground. The boy is building a castle in a sandbox. The mother is sitting on a bench and she watches her son while holding a book in her hands. I, the agent, am walking by and observe the scene. Clearly, the scene is a part of reality and the agent manages to individuate the situation by observation but he holds only a limited part of the information that is theoretically available. The part consists of objects, e.g. the boy and the book, having properties, e.g. the age of the boy. Objects stand in relation to one another, e.g. the parent is the mother of the boy and the castle is made of sand. Thus, the part is structured. The objects are located in space-time. The agent is using language to convey information he holds about the situation.

2.3.1 Main Primitives

This section briefly presents some of the main primitives of the mathematical ontology of situation theory, following the notation by Devlin (1991). The mathematical ontology provides us with a formal definition of situation. The illustrative examples are drawn from the domain of crop disease pressure situation modelling.

Infon As described above, situations are the limited parts of the world about which agents exchange information. It is informational items, called *infons*, that form such information.

Definition 2.3.2 (Infon)**.** An infon σ is formally defined as the $(m+2)$-tuple

$$\ll R, a_1, \ldots, a_m, i \gg$$

where R is an n-place relation; $a_1, \ldots, a_m$ $(m \leq n)$ are objects appropriate for the argument places $i_1, \ldots, i_m$ of R; and $i = 0, 1$ is the polarity of the infon, i.e. its truth value. The objects $a_1, \ldots, a_m$ represent the basic informational unit of the infon. They stand in relation R if $i = 1$ and do not stand in relation R if $i = 0$. The infon is unsaturated if $m < n$ and saturated if $m = n$. An infon of a situation that actually occurred in the world is called a fact.

Example 2.3.2 (Infon)**.** The *outbreak*-relation infon

$$\ll \text{outbreak, Pyrenophora teres, Barley, August 2014, Pelto, } 1 \gg$$

is the informational item for an outbreak of *Pyrenophora teres* (the pathogen) in barley (the crop) occurred during the entire month of August 2014 (the temporal location) at the Pelto agricultural parcel (the spatial location). The objects `Pyrenophora teres`, `Barley`, `August 2014`, and `Pelto` represent the informational unit of the infon. They stand in the *outbreak* relation. As the outbreak did occur in the world, the infon is called a fact.

Situation Situations are members of the mathematical ontology of situation theory, just like infons.

Definition 2.3.3 (Situation, formal)**.** Let s be a situation. Given an infon σ, the relation

$$s \models \sigma$$

is read s *supports* σ, and expresses that the infon σ is made factual by the situation s. In other words, σ is an item of information that is true of s. A situation $s \models \Sigma$ supports a set of infons Σ if s supports every infon $\sigma \in \Sigma$.

Minimality Condition The relation R is an abstract object. Among other elements, its structure holds a collection of minimality conditions. Minimality conditions "determine which particular groups of argument roles need to be filled in order to produce an infon." An infon that satisfies the minimality conditions is a *well-defined* infon.

Example 2.3.3 (Well-defined infon)**.** In our example, the fact that an agricultural parcel covers a single crop group could motivate a minimality condition that specifies that a well-defined infon with *outbreak* relation must have objects for pathogen, temporal location, and spatial location. An object for the crop is thus not required. The infon

$$\ll \text{outbreak}, \text{Pyrenophora teres}, \text{August 2014}, \text{Pelto}, 1 \gg$$

satisfies the minimality conditions of the *outbreak* relation. The infon is thus well-defined.

Argument Role In addition to a collection of minimality conditions, the structure of the abstract object R also holds a fixed collection of argument roles of R. Each argument role has an associated type that determines "the type of object that may legitimately fill that argument role." The expression $x : T$ is used to state that the object x is of type T. Situation theory provides for various basic types, e.g. the type SIT for situations and the type REL^n for n-place relations. The expression $role \rightsquigarrow a$ states that a particular object a fills the argument role

role. An infon with argument roles filled with objects of appropriate type is a *well-formed* infon.

Example 2.3.4 (Argument roles)**.** For our *outbreak* relation, the fixed collection of argument roles can be specified as

$$\langle \text{outbreak} \mid \textit{pathogen}, \textit{crop}, \textit{duration}, \textit{parcel} \rangle$$

where *pathogen*, *crop*, *duration*, and *parcel* are the argument roles of the *outbreak* relation. We may specify argument role type as

$$\langle \text{outbreak} \mid \textit{pathogen:P}, \textit{crop:C}, \textit{duration:TIM}, \textit{parcel:LOC} \rangle$$

where P is the type of all pathogens, C is the type of all crops, TIM is the type of all temporal locations, and LOC is the type of all spatial locations. The infon in Example 2.3.3 may also be written with the following, or any other rearrangement of, filled argument roles

$$\ll \text{outbreak}, \textit{pathogen} \rightsquigarrow \text{Pyrenophora teres}, \textit{crop} \rightsquigarrow \text{Barley}, \\ \textit{duration} \rightsquigarrow \text{August 2014}, \textit{parcel} \rightsquigarrow \text{Pelto}, 1 \gg$$

where `Pyrenophora teres` : P, `Barley` : C, `August 2014` : TIM, and `Pelto` : LOC. The infon is saturated.

Parameter In addition to objects of appropriate type, an argument role can also be filled with a parameter that makes reference to *arbitrary* objects of appropriate type. The notation $\dot{a}$ is used to denote parameters and contrast the parameter from the object a. As parameters make reference to arbitrary objects of a given type, a mechanism, called *anchor*, is introduced to assign objects to parameters. The specific character used for a parameter can reflect the type of anchored objects. For instance, the parameters $\dot{t}$ and $\dot{l}$ generally anchor objects of type TIM and LOC, respectively.

Example 2.3.5 (Parameter)**.** The *outbreak*-relation infon

$$\ll \text{outbreak}, \dot{p}, \text{August 2014}, \text{Pelto}, 1 \gg$$

with filled argument roles and different arrangement written as

$$\ll \text{outbreak}, \textit{duration} \rightsquigarrow \text{August 2014}, \textit{pathogen} \rightsquigarrow \dot{p}, \textit{parcel} \rightsquigarrow \text{Pelto}, 1 \gg$$

includes the parameter $\dot{p}$. It stands for an arbitrary pathogen. The parameter $\dot{p}$ may anchor an object $p : P$, such as `Pyrenophora teres` : P.

In addition to the parameter for arbitrary pathogens, the *outbreak*-relation infon

$$\ll \text{outbreak}, \dot{p}, \dot{t}, \dot{l}, 1 \gg$$

includes parameters $\dot{t}$ and $\dot{l}$ to make reference to arbitrary objects of type temporal location and spatial location, respectively.

The presented primitives of the mathematical ontology of situation theory provide us with an understanding for how information about situations is modelled in situation theory. This section introduced only a fraction of situation theory as it is presented in full length by Devlin (1991), and summarized by Devlin (2006). However, the presentation covers the aspects required in this dissertation. Indeed, of primary interest are the relation between situation and infon, the infon itself, parameters, and the anchor mechanism.

2.3.2 Situation in Environmental Monitoring

The notion of situation, as defined in situation theory, is arguably of interest to the modelling of information about environments monitored by systems. Assume the environmental monitoring system to be the agent. The monitored environment is a part of reality. It contains objects, in particular environmental phenomena. The objects stand in relation to one another. The part of reality is thus structured. The agent has a narrow view on reality, largely delimited by the scope of monitoring which determines, e.g., what precisely is monitored where and when, and for what purpose. The scope of monitoring also practically limits the information such agent has about the monitored environment.

Example 2.3.6. Assume the agent to be an environmental monitoring system for the prediction of crop disease pressure in agriculture. Concretely, the system includes an environmental sensor network that monitors certain weather parameters, such as ambient air temperature, and computes the outbreak risk for certain pathogens in the crop of an agricultural parcel, located where the weather parameters are observed by the environmental sensor network. The agricultural parcel and the local weather form the individuated structured part of reality. The view on reality is spatially delimited by the polygon boundary of the agricultural parcel and a limited volume of the atmosphere. The goal of monitoring determines which weather parameters are monitored as well as what information the agent obtains and maintains about the objects and relations among them. Objects include the crop and pathogens, such as barley and *Pyrenophora teres*, respectively. They may stand in relation as a pathogen can, for instance, affect a crop.

We suggest that environmental monitoring systems observe situations, and that the concept can serve as unifying abstraction for knowledge about the monitored environment that such agents can obtain, maintain, and share. An environmental monitoring system is different from, and should not be confused with, environmental sensor networks, which typically observe the signal of *properties* of environmental phenomena in space-time, rather than situations.

The following section introduces the concept of ontology, as understood in information science. Ontology and related technologies enable the explicit, formal, representation of situational knowledge in environmental monitoring systems. In systems, situation is thus a knowledge object. The section also introduces the core ontologies and related technologies adopted in this work.

2.4 ONTOLOGY

Borrowed from philosophy, the term ontology has recently received considerable attention in various computational fields of study (Gruber, 1993; Guarino and Giaretta, 1995; Studer et al., 1998). Gruber (1993) defines ontology as:

Definition 2.4.1 (Ontology, Gruber)**.** Ontology is an explicit specification of a conceptualization.

For two decades, this definition, and variants thereof, has been prevalent in the literature. Of particular focus in this section is the analysis of Gruber's definition developed by Guarino et al. (2009). The following paragraphs first briefly summarize Guarino et al.'s analysis of the notions of conceptualization and explicit specification, and then discuss definitions that extend Gruber's definition before presenting some alternative definitions.

2.4.1 Definition Analysis

We briefly discuss the notions of conceptualization and explicit specification.

Conceptualization The notion of conceptualization dates back to Genesereth and Nilsson (1987) who formally define conceptualization as:

Definition 2.4.2 (Conceptualization, Genesereth and Nilsson)**.** A conceptualization is the triple consisting of a universe of discourse, a functional basis set for that universe of discourse, and a relational basis set.

Genesereth and Nilsson explicate that a universe of discourse is the "set of objects about which knowledge is being expressed." Furthermore, the authors state that "an object can be anything about which we want to say something." Functions and relations are two kinds of "interrelationship among the objects in

a universe of discourse." A functional (relational) basis set includes the functions (relations) that are "emphasized in a conceptualization," of all those that are possible. Importantly, the entities of a conceptualization according to Genesereth and Nilsson are *extensional*. Therefore, the objects of a conceptualization are enumerated, and interrelationships are sets of objects. It follows that a conceptualization describes a particular state of affairs, a particular world (Guarino and Giaretta, 1995).

Guarino et al. (2009) reformulates and simplifies Genesereth and Nilsson's mathematical representation of a conceptualization as:

Definition 2.4.3 (Conceptualization, Guarino et al.)**.** A conceptualization is the tuple $(\mathcal{D}, \mathscr{R})$ consisting of a universe of discourse $\mathcal{D}$ and a set $\mathscr{R}$ of relations on $\mathcal{D}$.

The tuple is an extensional relational structure and is equivalent to Genesereth and Nilsson's notion of conceptualization. The universe of discourse $\mathcal{D}$ is a set of objects and $\mathscr{R}$ is a set of extensional relations on $\mathcal{D}$ (i.e. sets of ordered tuples of elements of $\mathcal{D}$). Thus, both Genesereth and Nilsson's conceptualization and Guarino et al.'s extensional relational structure reflect a specific state of affairs.

Example 2.4.1 (Extensional relational structure)**.** Farmers grow cereal crops at agricultural parcels. The crops may be affected by fungal pathogens. Our universe of discourse $\mathcal{D}$ contains the parcels, crops, pathogens, each identified by a code. The set $\mathscr{R}$ contains the unary relations Parcel, Crop, and Pathogen, as well as the binary relations grown-at and affected-by. The corresponding extensional relational structure $(\mathcal{D}, \mathscr{R})$ is:

- $\mathcal{D} = \{ap01, ap02, \ldots, cc01, \ldots, fp01, \ldots\}$
- $\mathscr{R} = \{Parcel, Crop, Pathogen, grown\text{-}at, affected\text{-}by\}$

Relation extensions reflect a specific state of affairs, a specific world. The universe $\mathcal{D}$ consists of all parcels, crops, and pathogens. The binary relations grown-at and affected-by are sets of tuples. In our world, the crop *cc01* is grown-at the parcel *ap02* and is affected by the pathogen *fp01*.

- $Parcel \cup Crop \cup Pathogen = \mathcal{D}$
- $Parcel = \{ap01, ap02, \ldots\}$
- $Crop = \{cc01, \ldots\}$
- $Pathogen = \{fp01, \ldots\}$
- $grown\text{-}at = \{\ldots, (cc01, ap02), \ldots\}$
- $affected\text{-}by = \{\ldots, (cc01, fp01), \ldots\}$

Guarino et al. highlight that the extensional notion of conceptualization is problematic in a definition for ontology, "mainly because it depends too much on a specific state of the world." Arguably, adding a tuple to an extensional relation of a conceptualization should not result into a different conceptualization. As Guarino et al. argue, a conceptualization "is about concepts" and "should not change when the world changes."

Based on this observation, Guarino et al. propose the *intensional* relational structure $(\mathcal{D}, \mathcal{W}, \mathcal{R})$ as mathematical representation of a conceptualization. $\mathcal{W}$ is a set of possible worlds and $\mathcal{R}$ is a set of intensional relations on $< \mathcal{D}, \mathcal{W} >$. An intensional relation of arity n on $< \mathcal{D}, \mathcal{W} >$ is a total function from the set $\mathcal{W}$ into the set of all n-ary extensional relations on $\mathcal{D}$. In contrast to the extensional relational structure, the intensional relational structure allows for different state of affairs (worlds) to be described by a single conceptualization. As Guarino et al. argue, it is thus more adequate for a definition of ontology.

Example 2.4.2 (Intensional relational structure)**.** We demonstrate how the intensional relational structure supports the description of different state of affairs using our example for agricultural parcels at which farmers grow cereal crops, which may be affected by fungal pathogens. The intensional relational structure $(\mathcal{D}, \mathcal{W}, \mathcal{R})$ is:

- $\mathcal{D} = \{ap01, ap02, \ldots, cc01, \ldots, fp01, \ldots\}$
- $\mathcal{W} = \{w_1, w_2, \ldots\}$
- $\mathcal{R} = \{Parcel^1, Crop^1, Pathogen^1, grown\text{-}at^2, affected\text{-}by^2\}$

The intensional relations may thus map to different extensions in different worlds, as shown for the two binary relations:

- for all worlds w in $\mathcal{W}$: $Parcel^1(w) \cup Crop^1(w) \cup Pathogen^1(w) = \mathcal{D}$
- for all worlds w in $\mathcal{W}$: $Parcel^1(w) = \{ap01, ap02, \ldots\}$
- for all worlds w in $\mathcal{W}$: $Crop^1(w) = \{cc01, \ldots\}$
- for all worlds w in $\mathcal{W}$: $Pathogen^1(w) = \{fp01, \ldots\}$
- $grown\text{-}at^2(w_1) = \{\ldots, (cc01, ap02), \ldots\}$
- $grown\text{-}at^2(w_2) = \{\ldots, (cc02, ap02), \ldots\}$
- $grown\text{-}at^2(w_3) = \ldots$
- $affected\text{-}by^2(w_1) = \{\ldots, (cc01, fp01), \ldots\}$
- $affected\text{-}by^2(w_2) = \ldots$

Explicit Specification Having discussed the notion of conceptualization, we now return to Gruber's definition of ontology. The *explicit specification* of a conceptualization rests on a language, one that enables reference to the elements of a conceptualization—in other words, one that enables us to talk about a conceptualization.

Of particular interest are logical languages, with vocabulary consisting of a set of constant and predicate symbols. The symbols in the vocabulary of a language obtain meaning through interpretations. Each symbol has both extensional and intensional interpretations. Thus, explicit specification can occur extensionally or intensionally.

However, extensional specification is impossible in most cases, and otherwise impractical because it requires listing the extensions of every relation for all possible worlds. More effective is intensional specification, which occurs by means of axioms that constrain the possible interpretations for the symbols.

An ontology is a set of axioms. Guarino et al. (2009) underscore that an ontology is, strictly speaking, an *approximate* specification of a conceptualization, in other words a partial account of a conceptualization. This is because the degree of specification depends on various factors, e.g. the purpose of the specification (Guarino and Giaretta, 1995).

2.4.2 Alternative Definitions

Building on Gruber's definition, Borst (1997) defines ontology as a "formal specification of a shared conceptualization." In this definition, the specification must be formal, i.e. machine readable (Guarino et al., 2009). Formal languages such as logical languages meet this requirement, while natural language does not. Furthermore, the conceptualization must be shared, i.e. it must reflect a consensus among ontology stakeholders. Indeed, specifications of a conceptualization that lack consensus are arguably hard to reuse and are thus considered useless (Borst, 1997; Guarino et al., 2009). Studer et al. (1998) merge these definitions and define ontology as a "formal, explicit specification of a shared conceptualization."

Alternative definitions of ontology have been proposed in the literature by various authors. According to Neches et al. (1991) "an ontology defines the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vocabulary." According to Swartout et al. (1996), an ontology is "a hierarchically structured set of terms for describing a domain that can be used as a skeletal foundation for a knowledge base." Thus, Swartout et al. explicate, "an ontology provides a skeletal structure for a knowledge base." According to Hendler (2001), an ontology is "a set of knowledge terms, including the vocabulary, the semantic interconnections, and some simple rules of inference and logic for some particular topic." Hendler notes that this definition reflects (or reflected, at the time of his writing)

how the term ontology is used within the semantic web community. Berners-Lee et al. (2001), and co-author Hendler, note that to artificial intelligence and web researchers "an ontology is a document or file that formally defines the relations among terms."

Some elements of these alternative definitions—such as term, vocabulary, interconnection, and domain—are reminiscent of Gruber's definition, while others are new. Swartout et al. introduce the notion of knowledge base as an entity distinct from ontology. Swartout and Tate (1999) clarify that a knowledge base uses the set of terms provided by an ontology "to represent what is true about some real or hypothetical world." By introducing the notion of rule, Hendler's definition acquires reasoning as additional characteristic.

2.4.3 Semantic Web Technologies

The recent popularity of ontology in information science can arguably be attributed to the activities within the semantic web initiative of the World Wide Web Consortium (W3C). Berners-Lee et al. (2001) had already considered ontology to be a basic component of the semantic web, and envisioned that ontologies would enable computers to "manipulate [...] terms much more effectively in ways that are useful and meaningful to the human user."

The idea underlying the semantic web, namely to evolve "objects from being principally human-readable documents to contain more machine-oriented semantic information" (Berners-Lee et al., 1994), had already been formulated in the early days of the web. The ideas gained momentum with the web's growth rate during the first decade of its existence. The amount of data made it "increasingly difficult to locate, organize, and integrate the available information" and increasingly obvious that computers needed to do better at these tasks (Heflin and Hendler, 2001). As computers had not succeeded in processing natural language, researchers sought to make the web more understandable to computers by giving data well-defined meaning (Berners-Lee et al., 2001).

Reaching what has been called the web's 'full potential' required, however, the development of several new technologies, including standards and tools. Specifically, the standardization of the syntactic form of data achieved with the Extensible Markup Language (Bray et al., 1998, XML) was to be attained also for the *semantic* content of data (Decker et al., 2000).

The following paragraphs briefly present the standards, more accurately W3C recommendations, that are of particular interest in this dissertation, namely the Resource Description Framework, RDF Schema, the Web Ontology Language, and the SPARQL Protocol and RDF Query Language.

Resource Description Framework (RDF) The Resource Description Framework (Lassila and Swick, 1999; Klyne and Carroll, 2004; Cyganiak et al., 2014b)

is a model of metadata, specifically a model of data about web resources (Lassila and Swick, 1999). Following Lassila and Swick (1999), the model consists of three object types, namely resources, properties, and statements. A *resource* is primarily a *web* resource, such as a web page or an image linked to a web page. However, resources do not need to be accessible on the web. A resource may thus be a physical object, such as a sensing device, or an abstract concept. Generally, any entity that can be named by a Uniform Resource Identifier (Berners-Lee et al., 2005, URI) is a resource and a member of the set of RDF resources. A *property* is a specific "relation used to describe a resource" and is a member of the set `rdf:Property`, which is a subset of the set of RDF resources. A *statement* is a triple consisting of a resource, a property, and the value for the property of the resource. Statements are members of the set `rdf:Statement`. The three elements of the triple are called, respectively, the subject, the predicate, and the object of the RDF statement. The object of a statement can be a resource or a literal. A literal is a value of primitive data type, in particular XML data type, and is a member of the set of RDF literals.

Typically, an RDF statement is represented graphically as two nodes and a directed arc. The two nodes are for the subject and the object of the statement, respectively. The arc is for the property, and is directed from the subject to the object. Text representations of a statement include <s, p, o> and p(s, o), whereby s, p, o stand for subject, predicate, and object, respectively. To exchange RDF statements, in particular between computers, Lassila and Swick (1999) had proposed an XML syntax for RDF, which Beckett (2004) later on revised. Various other RDF syntaxes have been developed, such as Turtle (Prud'hommeaux and Carothers, 2014) and N-TRIPLES (Carothers and Seaborne, 2014).

RDF specifies three additional important features. First, RDF defines the property `rdf:type`, member of `rdf:Property`. The `rdf:type` property enables primitive typing in RDF. RDF requires that the subject and the object in a statement with `rdf:type` predicate are members of the set of RDF resources. Second, RDF defines three types of container objects, namely bag, sequence, and alternative. RDF containers refer to collections of resources or literals. Third, RDF supports a mechanism, called reification, that enables making statements about other RDF statements. Given the statement <s, p, o> , the reified statement is a resource with the following four properties: `rdf:subject` with value s, `rdf:predicate` with value p, `rdf:object` with value o, and `rdf:type` with value `rdf:Statement`. This resource may have further properties.

RDF Schema (RDFS) In RDF, the property of a statement represents a relationship between two resources. RDF does not provide a mechanism to *describe* such relationships, for instance describe the particular groups of resources a property relates. This is addressed by RDF Schema (Brickley and Guha, 2004, 2014). RDFS

is a data-modelling vocabulary for RDF data. The RDFS vocabulary consists of particular sets of resources.

Any entity described in RDF is a resource, and instance of `rdfs:Resource`. Thus, `rdfs:Resource` includes "everything." Resources may be divided into groups, i.e. classes. All classes are therefore subclasses of `rdfs:Resource`. Among classes, `rdfs:Class` is the class of *resources* that are RDF classes. Clearly, `rdfs:Class` is a subclass of `rdfs:Resource`. A class, including `rdfs:Resource`, is an instance of `rdfs:Class`. Being the class of all RDF properties, `rdf:Property` is an instance of `rdfs:Class`. RDFS defines `rdfs:Literal`, the class of all RDF literals. Being a class, `rdfs:Literal` is an instance of `rdfs:Class` and a subclass of `rdfs:Resource`.

In addition to classes, RDFS defines a particular set of properties, which are instances of the class `rdf:Property`. Specifically, the property `rdfs:subClassOf` enables the construction of class hierarchies. The statement

<C, rdfs:subClassOf, D>

states that all instances of C are instances of D, and that C, D are instances of `rdfs:Class`. On a similar line, the property `rdfs:subPropertyOf` enables the construction of property hierarchies. Given the statement

<P, rdfs:subPropertyOf, Q>

pairs of resources related by P are also related by Q. The statement also implies that P, Q are instances of `rdf:Property`.

Two further properties defined by RDFS are of particular interest, namely `rdfs:domain` and `rdfs:range`. The statements

<P, rdfs:domain, C> <P, rdfs:range, D>

state that for statements <r, P, s> the resources r and s are instances of the classes C and D, respectively, that P is an instance of `rdf:Property`, and that C, D are instances of `rdfs:Class`.

Web Ontology Language (OWL) RDFS supports the construction of basic ontologies. The construction of ontologies with richer semantics is supported by the Web Ontology Language (Bechhofer et al., 2004; Motik et al., 2012). A Web Ontology Language (OWL) ontology consists of a set of axioms and, typically, a set of assertions (i.e. "facts about individuals"). The set of axioms consists of class axioms and property axioms. The set of assertions consists of concept and role assertions, i.e. class membership and property values of individuals. The following paragraphs describe the core features of OWL, in particular how the language supports the definition of axioms and assertions.

OWL supports the description of classes by means of six types of so-called class descriptions. An `owl:Class`, which is defined as a subclass of `rdfs:Class`, can be described through (1) a class name (as in RDFS); (2) an exhaustive enumeration of individuals, instances of the described class; (3) a property restriction; the (4) intersection or (5) union of two or more class descriptions; or (6) the complement of a class description. A property restriction describes the class of all individuals that satisfy the restriction. There exist two types of property restrictions: value constraints and cardinality constraints. A value constraint restricts the range of the property when applied to the particular class description (which is thus different from `rdfs:range`). This type of property restriction includes constraints analogous to universal and existential quantifiers of Predicate logic. A cardinality constraint restricts the number of values a property can take in the context of the particular class description. An instance of a class may have an arbitrary number of values for a particular property. Cardinality constraints can make a property required, allow only a specific number of values for a property, or specify that a property must not occur.

In addition to `rdfs:subClassOf`, inherited from RDFS, OWL introduces two further constructs for the definition of class axioms, i.e. `owl:equivalentClass` and `owl:disjointWith`. The building blocks of class axioms are class descriptions. It is a class *description* that is a subclass of another class description. Subclass axioms represent necessary conditions for establishing class membership of an individual. In contrast, equivalent class axioms represent necessary *and* sufficient conditions.

OWL distinguishes between object and data type properties. An object property is an instance of the class `owl:ObjectProperty` and relates two individuals. A data type property is an instance of the class `owl:DatatypeProperty` and relates an individual and a literal. Both are subclasses of `rdf:Property`. OWL defines several constructs for property axioms in addition to those inherited from RDFS, such `owl:equivalentProperty` and `owl:TransitiveProperty`.

Facts about individuals are defined in OWL with axioms about individuals (assertions). Of particular interest are axioms that specify the class membership of an individual (concept assertions) and axioms that specify the property values of individuals (role assertions). For instance, given the class description C and the individual a, C(a) is the concept assertion for stating that a is an individual instance of C. Given the property P, P(a, b) is the role assertion for stating that the individuals a and b are related by P. Additionally, OWL also supports stating that two individuals are same or are different.

SPARQL Protocol and RDF Query Language (SPARQL) Having provided an overview of RDF, RDFS, and OWL we now turn to the SPARQL Protocol and RDF Query Language (Prud'hommeaux and Seaborne, 2008; Harris and Seaborne, 2013). SPARQL is of interest here as query language for RDF.

The core SPARQL construct is arguably the *triple pattern*. A triple pattern is like an RDF triple except that the subject, predicate, and object may each be a variable. A triple pattern *matches* RDF triples when variables can be substituted. For instance, given a set of RDF triples, upon which SPARQL queries are evaluated, a triple pattern with variables in the subject, predict, and object position matches all triples in the set. In SPARQL, a set of triple patterns is called a *basic graph pattern*. As RDF statements span a directed graph, a basic graph pattern matches an RDF sub graph.

SPARQL is reminiscent of the Structured Query Language (Chamberlin and Boyce, 1974, SQL). In fact, the declarative query language reuses several of the well-known SQL clauses, including `SELECT`, `WHERE`, and `ORDER BY`. A (group) graph pattern specifies the `WHERE` clause. Filters can be declared in order to restrict the solutions of a graph pattern according to a filter expression. Parts of the graph pattern may be optional, which is useful when sub graphs have irregular structure. SPARQL supports query forms other than `SELECT`. Of particular interest is the `CONSTRUCT` query form. The `SELECT` query form returns variables and their bindings. In contrast, the `CONSTRUCT` query form returns an RDF graph specified by a graph template. The `CONSTRUCT` query form is often useful in applications because the returned RDF graph can be further processed with RDF tools, including a SPARQL query engine.

2.4.4 Relevant Ontologies

This section introduces the core ontologies we adopt in this work. These are the Semantic Sensor Network ontology (Compton et al., 2012), the RDF Data Cube Vocabulary (Cyganiak et al., 2014a), the Situation Theory Ontology (Kokar et al., 2009), GeoSPARQL (Perry and Herring, 2012), OWL-Time (Hobbs and Pan, 2006), and the PROV ontology (Lebo et al., 2013). They form an ontological framework for situation-aware environmental monitoring systems.

Semantic Sensor Network Ontology (SSN) The semantic sensor network ontology is designed to "describe the capabilities and properties of sensors, the act of sensing and the resulting observations" (Compton et al., 2012). The SSN ontology aims at providing semantic interoperability of sensor data, on top of syntactic interoperability addressed in particular also by Open Geospatial Consortium (OGC) standards such as SensorML (Botts and Robin, 2007) and Observations and Measurements (Cox, 2011, O&M).

Though descriptions about the capabilities and properties of sensors are useful in applications, of most interest here are the observations resulting in the act of sensing, i.e. the observation perspective of the SSN ontology. To model observations, the SSN ontology defines the class `ssn:Observation`. Closely aligned with OGC standards and modelling of observations, an SSN observation is *for*

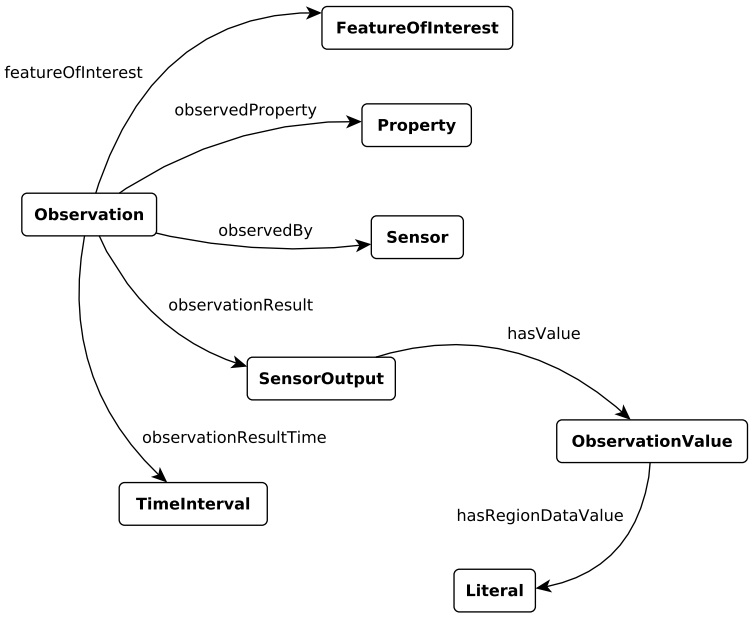


Figure 2.1: Relations between SSN observation and the sensor that made the observation, the observed property of the feature, the sensor output and observation value, and the time at which the observation was made.

a particular property *of* a feature and is *observed* by a sensor that implements some sensing method. Sensor is understood broadly to include physical devices as well as other entities that can implement a sensing method to observe a property, such as computational methods or laboratory set-ups. Naturally, in addition to descriptions for what was sensed, what made the observation and how it was made, SSN observations also describe the sensor output, which is often a numerical observation value. Finally, SSN observations can describe other metadata, in particular spatio-temporal data for where and when the observation was made. Ontological modelling of time and space are, however, not part of the SSN ontology. Figure 2.1 provides a graphical overview of the main relations between SSN observation and sensor, property, feature, observation value, and time.

RDF Data Cube Vocabulary (QB) The RDF data cube vocabulary is designed to represent multi-dimensional data in RDF. Fundamental to multi-dimensional data "is a set of observed values organized along a group of dimensions, together with associated metadata" (Cyganiak et al., 2014a). Observed values are modelled as `qb:Observation`. A QB observation relates to a `qb:DataSet`, which is thus a collection of observations. Datasets are generally structured. Accordingly, QB supports the definition of structures as `qb:DataStructureDefinition`.

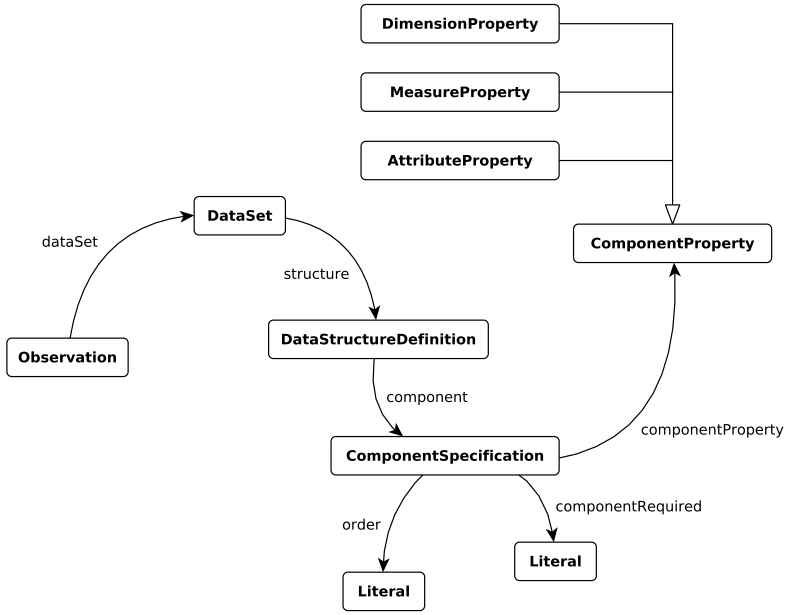


Figure 2.2: Relation between QB observation and the dataset with its data structure definition consisting of a set of component specifications. Component properties are RDF properties available to observations to relate property values.

One or more datasets may relate to the same data structure definition. A structure is described by a set of `qb:ComponentSpecification`. A component specification determines the `qb:ComponentProperty` as well as other metadata about the component, such as whether or not it is required and its order within the structure. QB supports three types of component properties, namely dimension, measure, and attribute properties. Component properties are RDF properties and are used to relate observations with values. Figure 2.2 provides a graphical overview of the relation between QB observation and dataset with data structure definition.

As an example, consider a typical comma-separated values file consisting of n labels on the first line and m lines with n numbers starting on the second line and ending on line $m+1$ of the file. The first line of the file can be translated into a QB data structure definition. Each of the n labels is translated to a component specification. The label itself maps to a component property while the position of the label in the list determines the value of the order property in the component specification. The m lines of the file form a $m \times n$ multi-dimensional dataset. This dataset relates to the described data structure definition. Each of the $2 \ldots m+1$ lines in the file can be translated into a QB observation. Each line consists of n numbers. The QB observation relates thus to the dataset and to the n numbers via the component properties as defined by the data structure definition.

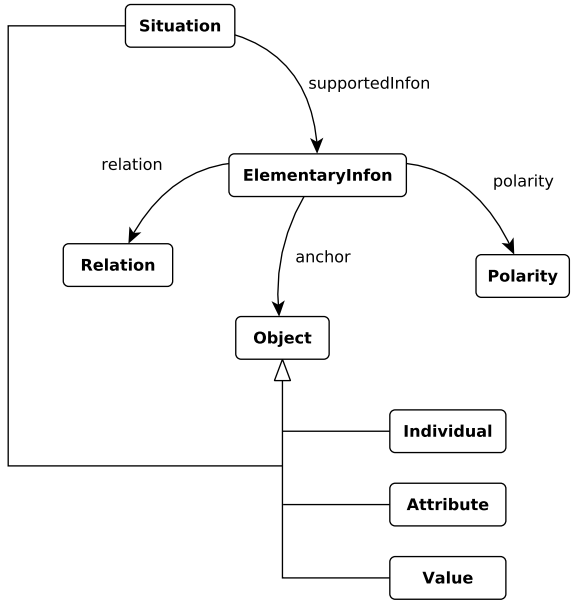
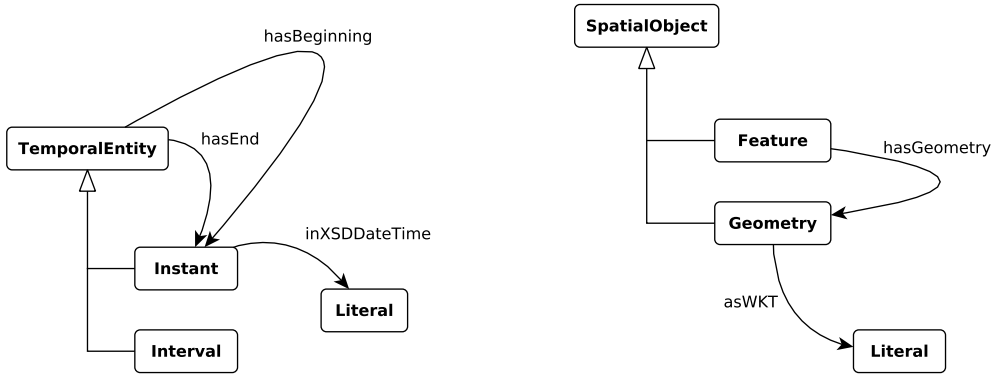


Figure 2.3: Relations between STO situation and supported infons with relation, objects, and polarity. Objects may be specialized as individuals, attributes, or values. Situations may also be objects in situations.

Situation Theory Ontology (STO) The situation theory ontology captures the situation theory of Barwise and Perry and Devlin "in terms of an OWL ontology [to allow] one to express situations in a commonly supported language with computer processable semantics" (Kokar et al., 2009). We have presented situation theory in Section 2.3. Naturally, the central STO class is `sto:Situation`. Individual situations occurring in the world are its instances. As STO follows situation theory, a situation supports one or more `sto:ElementaryInfon`. An elementary infon is a tuple consisting of a `sto:Relation`, a set of `sto:Object`, and a `sto:Polarity`. Objects may be, for instance, `sto:Individual` or `sto:Attribute`. Individuals participate in situations and attributes, such as temporal and spatial locations, may be of individuals or infons. Objects are anchored to elementary infons. Note that situations are objects and may thus be anchored to elementary infons. Figure 2.3 provides a graphical overview of the relations between STO situation and infon with relation, object, and polarity.

OWL-Time and GeoSPARQL SSN, QB and STO do not support the representation of time and space. However, they provide for possible relations to the vocabulary of specialized ontologies for the modelling of time and space, such as OWL-Time and GeoSPARQL, respectively. For instance, STO defines the class `sto:Time` as attribute. By aligning this class with a corresponding class of a spe-



(a) Relations between OWL-Time temporal entity with beginning and end instant with XSD textual representation.

(b) Relations between GeoSPARQL feature and geometry with WKT textual representation.

Figure 2.4: Relevant OWL-Time and GeoSPARQL concepts and relations. These specialized ontologies provide terms for the representation of time and space in SSN observations, QB observations, and STO situations.

cialized ontology for the modelling of time, such as OWL-Time, individuals of `sto:Time` inherit the vocabulary and ontological modelling of the specialized ontology. The following paragraphs briefly present OWL-Time and GeoSPARQL as two possible specialized ontologies adopted in this work for the representation of time and space, respectively, in OWL.

OWL-Time defines the class `time:TemporalEntity`, as well as its subclasses `time:Instant` and `time:Interval`. It also defines the two object properties `time:hasBeginning` and `time:hasEnd` used to relate a temporal entity with an instant. Finally, it defines the data type property `time:inXSDDateTime` used to relate an instant with a literal of type `xsd:dateTime`. Beyond these basic classes and properties, OWL-Time allows for the explicit representation of temporal descriptions (e.g. durations) and topological relations (e.g. before). Figure 2.4(a) provides a graphical overview of the relations between OWL-Time temporal entity and the XSD textual representation of instants.

GeoSPARQL defines the class `geo:SpatialObject`, as well as its subclasses `geo:Feature` and `geo:Geometry`. It defines the object property `geo:hasGeometry` used to relate a feature with a geometry. Finally, it defines the data type property `geo:asWKT` used to relate a geometry with a literal of type `geo:wktLiteral` (a GeoSPARQL data type) to allow for text representation of geometries. Beyond these most relevant classes and properties, GeoSPARQL supports the explicit representation of topological relations, in particular also those of the Region Connection Calculus (Randell et al., 1992). Figure 2.4(b) provides a graphical overview of the relations between GeoSPARQL feature and geometry with WKT textual representation.

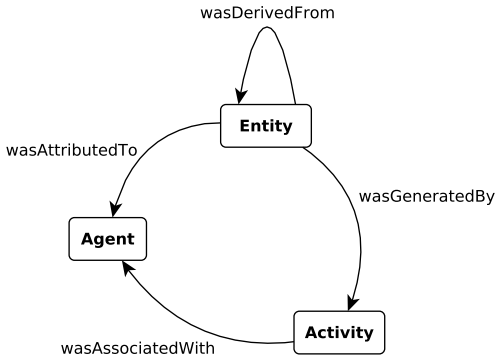


Figure 2.5: Relations between PROV entity, activity, and agent. The PROV ontology supports the representation of information about the provenance of SSN observations, QB datasets and observations, and STO objects in environmental monitoring systems, as well as information about the involved (software) agents and (algorithmic) activities.

Stocker et al. (2014) describe an ontology with a preliminary alignment of SSN, QB, STO, OWL-Time, and GeoSPARQL. The alignment consists primarily of axioms that relate key classes and properties of the ontologies. The resulting ontological framework is also extended with a few additional classes. Of primary interest is the introduction of the classes `SensorObservation` and `DatasetObservation`, which are defined to be equivalent to `ssn:Observation` and `qb:Observation`, respectively. The distinction is useful for human agents because the term 'observation' is used in both the SSN ontology and the QB vocabulary, and is thus ambiguous.

PROV Ontology (PROV-O) PROV is a specification for provenance designed for the representation of the origins of digital objects in form of descriptions about "the entities and activities involved in producing and delivering or otherwise influencing a given object" (Gil and Miles, 2013). In PROV, provenance is, generally, of entities, which can be physical, digital, or conceptual. Entities can be derived from other entities and they are generated by activities. Activities are the processes through which entities come into existence. Associated with activities are agents, which can be, e.g., persons or, of most interest here, software. Figure 2.5 provides a graphical overview of the relations between PROV entity, activity, and agent.

Stocker et al. (2015b) describe a preliminary alignment of PROV-O with the alignment presented in Stocker et al. (2014). The alignment consists primarily of axioms that model sensor observations, dataset observations and STO objects, including situations, as PROV-O entities; SSN sensors and software agents used in data processing and knowledge acquisition as PROV-O agents; and SSN stimuli and operations such as aggregation and classification as PROV-O activities.

2.5 MODELLING

This last section of Chapter 2 introduces models, specifically computational models. Modelling, in particular environmental modelling, is the "process of designing, building and using an environmental model" (Barnsley, 2007). Models play an important role in this dissertation because they enable the extraction of situational knowledge from data, as well as the processing of data, in environmental monitoring systems. Mulligan and Wainwright (2004) define model as:

Definition 2.5.1 (Model)**.** A model is an abstraction of a real system, it is a simplification in which only those components which are seen to be significant to the problem at hand are represented in the model.

Of particular interest here are (real) *environmental* systems, e.g. an agricultural field, and thus environmental models. Mulligan and Wainwright stress that a model is a simplification and has a purpose; it is expected to represent "a complex reality in the simplest way that is adequate for the purpose of the modelling." In other words, models ought to follow the principle of parsimony, which dictates that good models are those with good explanation or predictive power while having low parameter or process complexity.

Mulligan and Wainwright outline typical purposes of environmental models. For example, the purpose of a model may be to simulate and predict the behaviour of a system and thus to assist system understanding. Such a model can support research activities. Models can also be research products, "particularly when they can be used by others and thus either provide the basis for further research or act as a tool in practical environmental problem solving or consultancy." Especially when used by others, models can serve in communicating science or assisting decision-making processes.

2.5.1 Typology

A cursory read of the literature suggests that there is no agreement on a classification of models, a conclusion supported by Mulligan and Wainwright who state that "[t]here are no universally accepted typologies of models." The classification proposed by Mulligan and Wainwright first separates mathematical models from hardware models—a somewhat atypical distinction, yet arguably useful in environmental science.

Hardware, or physical, models are scaled-down versions of real systems and "give a degree of control on the systems under investigation." The advantage of hardware models rests in their ability to "couple the scientific rigour of observation with the controllability of mathematical modelling." However, hardware models can be expensive, are difficult to replicate, and are relatively uncommon. Furthermore, hardware models can represent a process only to the extent

to which it is understood and can be replicated. Finally, the model hardware can interfere with processes occurring in the system under investigation. An example given by Mulligan and Wainwright is the apparatus for free-air CO_2 enrichment experiments, which has been used for other atmospheric gases as well, such as O_3 (Uddling et al., 2010). Mulligan and Wainwright also consider smaller instruments to be hardware models, such as the Parkinson leaf chamber (Parkinson et al., 1980) used to measure the photosynthesis and transpiration in leaves.

In contrast to hardware models, mathematical models "represent states and rates of change according to formally expressed mathematical rules." Mulligan and Wainwright suggest five dimensions according to which mathematical models may be classified, namely the conceptual, integration, mathematical, spatial, and temporal. Each dimension distinguishes two or more categories, and the category for mixtures.

The conceptual dimension distinguishes three model classes, namely empirical, conceptual, and physically-based models. Empirical models "describe observed behaviour between variables on the basis of observations alone and say nothing of process." Typically, empirical models are mathematical functions that fit the observed relationship between variables. Empirical models "have high predictive power but low explanatory depth," meaning that they often agree with observations but cannot explain the process underneath an outcome. Conceptual models are empirical models that additionally describe observed behaviour "on the basis of preconceived notions of how the system works." Compared to pure empirical models, conceptual models have thus somewhat greater explanatory depth. Referring to Beven (2002), Mulligan and Wainwright state that physically-based models "should be derived deductively from established physical principles and produce results that are consistent with observations." However, Mulligan and Wainwright note that in reality "there is a continuum of models that falls broadly under the heading of physically-based, but that might include some level of empirical generalization." In contrast to empirical and thus conceptual models, physically-based models "tend to have good explanatory depth [but] are characterized by low predictive power."

The integration dimension distinguishes model equation integration, namely models with analytical solutions obtained by solving differential equations or models with numerical solutions obtained by solving difference equations. The mathematical dimension distinguishes deterministic and stochastic models. For a given input, models with deterministic equations always produce the same output. In contrast, in stochastic models a given input can produce different outputs. The spatial dimension distinguishes spatial types. In so-called lumped models a potentially spatially heterogeneous environment is modelled as a single value. Semi-distributed models may have multiple lumps. Distributed models break space into discrete regular or irregular units. The spatial dimension

also distinguishes one-/two-/tree-dimensional models. Finally, the temporal dimension distinguishes static models, which do not handle time, and dynamic models, which handle time explicitly.

Different classification schemes have been proposed to categorize environmental models. Robson (2014) argues that a "reasonably intuitive categorization is into two classes," i.e. mechanistic and statistical models. Robson cites Sharpe (1990) who proposes the additional class of theoretical mathematical models. Guisan and Zimmermann (2000) discuss the three model classes analytical, mechanistic, and empirical. How model classes are named varies among classification schemes. Analytical models are also called mathematical; mechanistic models are also called process-based, deterministic, physics-based, physiological, or causal; empirical models are also called statistical, phenomenological, or data based (Guisan and Zimmermann, 2000; Robson, 2014).

2.5.2 Relevant Models

Following the classification by Mulligan and Wainwright, of particular interest in this dissertation are mathematical models, rather than hardware models, specifically empirical models as well as physically-based empirical mixed models. Among the empirical models, this dissertation utilizes computational models in machine learning (Mitchell, 1997), in particular Multilayer Perceptron (MLP) artificial neural networks (Haykin, 1998).

An artificial neural network is a non-linear regression supervised learning model, "developed by training the network to represent the relationships and processes that are inherent within the data" (Solomatine et al., 2008). An MLP artificial neural network consists of a set of neurons that form the so-called input layer, one or more hidden layers, and the output layer of the network. MLP is trained in a supervised manner, i.e. by means of a labelled training dataset, using error back-propagation learning, which consists of a forward pass and a backward pass through the layers. In the forward pass, the signal resulting from the application of an input vector is propagated through the network in a forward direction and the actual response of the network at the output layer is recorded. In the backward pass, the recorded response of the network at the output layer is subtracted from a desired response (the label) to produce an error signal, which is propagated through the network in a backward direction. In the backward pass the network is adjusted in order to align the actual response with the desired response. An example for the application of MLP to a concrete problem can be found in Paper **II**. A summary of this application is provided next in order to elucidate how such models are utilized and thus clarify their primary role in this dissertation, namely to extract information from data.

Paper **II** describes the utilization of MLP artificial neural networks for the detection and classification of road vehicles. Vehicles are detected in road-

pavement vibration data obtained in measurement implemented by several accelerometer sensing devices installed at approximately 45 m relative distance on one side of a road section. Accelerometer sensing devices are said to "observe the vibration of the road pavement." Vehicles are sources of vibration and such vibration is detectable in data resulting from sensor measurement. Data in time domain is processed to vibration patterns in frequency domain. Vibration patterns are sampled and mapped to labels. Vibration patterns and corresponding labels form datasets used to train and validate MLP artificial neural networks for two *classification* tasks: vehicle detection and vehicle characterization. The aim in vehicle detection is to distinguish samples that include vehicle vibration from those that do not. Given detected vehicles, the aim in vehicle characterization is to distinguish samples for light vehicles from samples for heavy vehicles. For each classification task a distinct MLP artificial neural network is trained. Vibration patterns form the input and labels form the output of MLP artificial neural networks. In training, labelled samples are used to construct MLP artificial neural networks that support the automated mapping of vibration patterns to labels. In validation, trained networks are evaluated on (different) labelled samples to assess their classification performance. Validated MLP artificial neural networks are employed as models in a software system that classifies vibration patterns automatically and extracts information from road-pavement vibration data about vehicles that travel the monitored road section.

In addition to empirical models, of interest are also physically-based empirical mixed models. An example for the application of such model type to a concrete problem can be found in Paper **IV**. A summary of this application is provided next.

The assessment of disease pressure in agricultural crops is an important task in pest management. The application described in Paper **IV** utilizes a disease pressure model to compute daily risk values for pathogen and crop pairs. The model supports the computation of accumulated risks over space-time, and the assessment of (acute) outbreaks of pathogens in crops. Model input consists of static information about the crop and agricultural parcel, e.g. crop susceptibility and tillage, and dynamic weather data, e.g. temperature and wind speed. The core of the model is an equation for the computation of daily risk values. The equation combines estimates for the seasonal base risk and daily disease progress in the plant, spore development, spreading of disease, and infection probability. The ecological model is thus mechanistic and broadly falls into the class of physically-based models. We can further characterize the model as deterministic because a particular input will always produce the same output. Furthermore, the model is static as it does not explicitly handle time. However, the model is arguably not purely physically-based. Equation estimates, such as for spore development, are obtained via "table lookup," in other words by matching daily values for one or more dimensions against value ranges defined for

each dimension. For instance, spore development depends on temperature, in degrees Celsius, and the daily duration of leaf wetness, in hours. For both temperature and duration, the model defines value ranges, e.g. [10:15[and [8:10[for temperature and duration, respectively. Tables are pathogen specific and the values are obtained empirically. Therefore, the model is mixed physically-based empirical.

The two discussed models and applications are obviously different. However, they share the purpose, namely to enable the extraction of information from data. In the former case, data are for road-pavement vibration and information is for vehicles. In the latter case, data are for weather, crops and agricultural parcels, and information is for disease outbreaks in crops. In environmental monitoring systems, computational models can be the software agents that extract information from data, software agents that process data in order to enable information extraction, or software agents that process data and extract information.

2.6 SUMMARY

We have presented the central concepts underlying the dissertation. Environmental monitoring is the domain addressed by the research question: it is for environmental monitoring that we develop, implement in software, and validate on case studies a software process for the representation of situational knowledge acquired from data, in particular sensor data. The heterogeneous, often voluminous, and possibly real-time streamed data pose distinct challenges to automated data organization and interpretation in environmental monitoring. Furthermore, the complexity of data processing and knowledge acquisition, the heterogeneity of computational methods relevant to such processes, and the difficulty of integrating methods in machine learning and knowledge representation and reasoning pose additional challenges to automated data interpretation in environmental monitoring. The 'semantic gap' between data and knowledge is evidently a critical problem in environmental monitoring, and the need for suitable solutions only increases with increasing data volumes.

Situation awareness can inspire the modelling of situation-aware environmental monitoring systems as physical-socio-technical systems, whereby the monitored environment, the hardware and software, and people form the physical, technical, and social subsystems, respectively. As environmental monitoring systems are concerned (at least in this dissertation) with the perception, comprehension, and projection of environmental phenomena in space-time volumes, situation awareness—and the three levels model, in particular—can inspire the design of architectures for such systems.

In environmental monitoring systems, the notion of situation is a useful abstraction for information and knowledge relevant to comprehension, obtained from data acquired in perception, as well as knowledge resulting in projection.

Situation theory provides a formal definition of situation, and a structured object useful for the representation of situational knowledge in technical systems. Situational knowledge is obtained from data, and physically-based and data-driven models can support the automation of situational knowledge extraction from data. Ontology and related semantic technologies support the technical subsystems in the explicit representation of situational knowledge, and enable automated situational knowledge processing. We thus pursue a hybrid approach that attempts to make sense of data in environmental monitoring using both inductive and deductive techniques (Janowicz et al., 2015).

The following chapter presents the architecture and implementation of the proposed software framework for situation awareness in environmental monitoring. The framework is designed to support the implementation of software systems for the acquisition of situational knowledge from processed data acquired from environmental sensor networks, and the curation, access, and processing of situational knowledge.

3 Implementation

We present the architecture and implementation of the *Wavellite* software framework for situation awareness in environmental monitoring. Building on concepts presented in Chapter 2, Wavellite implements functionality common to applications presented in Chapter 4. The framework supports the development of environmental monitoring software systems that acquire situational knowledge from processed data collected from environmental sensor networks, as well as curate and process situational knowledge. The architectural description of Wavellite is aligned with the ENVRI Reference Model (ENVRI, 2013), which we extend with a model for knowledge acquisition from processed data, and the curation, access, and processing of knowledge. Section 3.1 briefly summarizes the ENVRI Reference Model. Section 3.2 summarizes the proposed reference model extension. The presentation is concise and only provides an overview. Both the ENVRI Reference Model and the extension are described in greater detail in Paper **V**. Having introduced the ENVRI Reference Model and the extension, Section 3.3 presents the Wavellite implementation.

3.1 REFERENCE MODEL

This section provides a brief overview of the ENVRI Reference Model. We summarize the model subsystems and discuss how the model is described from the science, information, and computational viewpoints.

The Common Operations of Environmental Research Infrastructures EU FP7 project (ENVRI) developed data and software components and services that are common to six environmental research infrastructures of the European Strategy Forum on Research Infrastructures (EU ESFRI) (Chen et al., 2013a,b), such as the Integrated Carbon Observation System (ICOS). Environmental research infrastructure are complex distributed systems that collect environmental (monitoring) data and manage such data for research. ENVRI aims at identifying common computational characteristics, develop an understanding of requirements, support and accelerate the construction of infrastructure, secure interoperability between infrastructures, avoid duplication of effort, and enable the reuse of resources and experiences.

The ENVRI Reference Model (ENVRI, 2013), hereafter ENVRI-RM, is arguably the primary result of the ENVRI project. ENVRI-RM is "a common ontological framework and standard for the description and characterisation of computational and storage infrastructures in order to achieve seamless interoperability between the heterogeneous resources of different infrastructures" (Chen

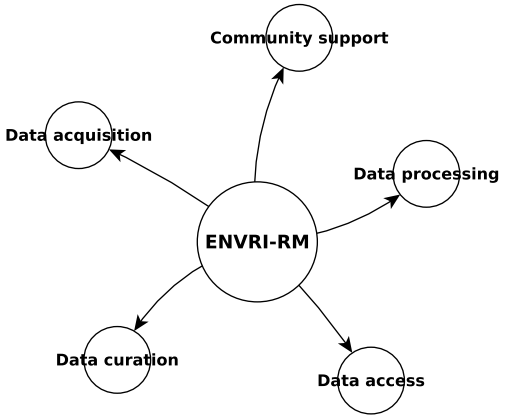


Figure 3.1: The five ENVRI-RM subsystems of environmental research infrastructure.

et al., 2013b). It provides "a universal reference framework for discussing many common technical challenges facing all of the ESFRI environmental research infrastructures" (Chen et al., 2013b). ENVRI-RM is publicly available and its latest version is V1.1 of August 30, 2013. Unless stated otherwise, this chapter quotes ENVRI-RM.

In analysing the six EU ESFRI environmental research infrastructures, the ENVRI project identified five key physical resources around which environmental research infrastructure "applications, services and software tools are designed and implemented" (Chen et al., 2013b). These are the sensor network, the storage, the (internet) communication network, application servers, and client devices.

Accordingly, ENVRI-RM divides the 'archetypical' environmental research infrastructure into five *subsystems*: data acquisition, data curation, data access, data processing, and community support. Figure 3.1 is a graphical overview of the five subsystems. A subsystem is "a set of capabilities that collectively are defined by a set of interfaces with corresponding operations that can be invoked by other subsystems" (Chen et al., 2013b). Functionality common to environmental research infrastructure is partitioned amongst the five subsystems. ENVRI-RM specifies a minimal model consisting of "fundamental functionality necessary to describe a functional environmental research infrastructure" (Chen et al., 2013b).

ENVRI-RM specifies environmental research infrastructure from three different *viewpoints*: science, information, and computational. Figure 3.2 is a graphical overview of the three viewpoints on environmental research infrastructure. A viewpoint on a system "is an abstraction that yields a specification of the whole system related to a particular set of concerns."

The science viewpoint "intends to capture the requirements for an environmental research infrastructure from the perspective of the people who perform

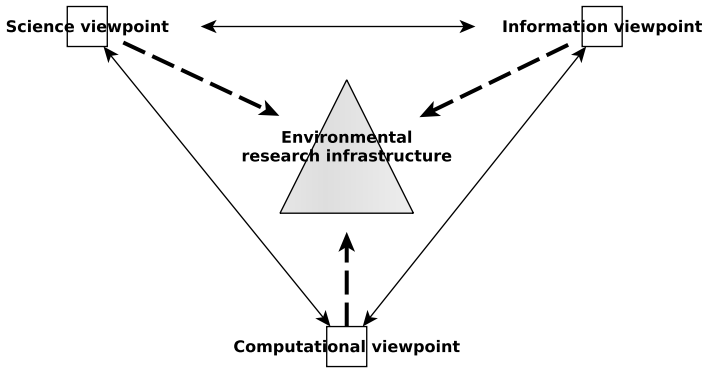


Figure 3.2: The three viewpoints from which ENVRI-RM specifies environmental research infrastructure.

their tasks and achieve their goals as mediated by the infrastructure." The viewpoint defines communities. Communities interact with the five subsystems to conduct scientific research. Each community is described for sets of relevant community roles and behaviours. A role in a community "is a prescribing behaviour that can be performed any number of times concurrently or successively." Roles can be active or passive. Active roles are associated with human actors. Passive roles are associated with non-human actors. A behaviour of a community "is a composition of actions performed by roles normally addressing separate [research activity] requirements." Behaviours are performed by roles and roles can address one or more requirements.

The information viewpoint provides "an abstract model for the shared information handled by the infrastructure." The viewpoint specifies the types of information objects and their relations. Furthermore, it "describes how the state of the data evolves as [a result] of computational operations" and "defines the constraints on the data and the rules governing the processing of such data." Information action types model how information is processed in the system, i.e. how a system manipulates information objects. Actions cause state changes in participating objects. Dynamic schemata "specify how the information evolves as the system operates, describing the allowable state changes of one or more information objects." In contrast, static schemata specify "instantaneous views of the information objects at a certain stage of the data life cycle." With subsystem schemata, the model also partitions information objects and information action types to subsystems.

The computational viewpoint specifies "the major computational objects expected within an [environmental] research infrastructure and the interfaces by which they can be interacted with." The archetypical environmental research infrastructure has a brokered service-oriented architecture. Functionality is thus

encapsulated by service objects, and access to service objects is managed by brokers. Access management includes, for instance, request validation. Service objects control resources. Interaction can be among computational objects within a subsystem or between subsystems.

3.2 REFERENCE MODEL EXTENSION

The ENVRI-RM focuses on data, their acquisition, curation, access, and processing. To model functionality for the acquisition of knowledge from data—and the curation, access, and processing of knowledge—we extend ENVRI-RM. The reference model *extension* is itself a model, and is called +K, which stands for 'plus knowledge'. The extension is inspired by ENVRI-RM, in the sense that it reuses some of the modelling choices made for ENVRI-RM. Furthermore, it can be superimposed on ENVRI-RM. The result of such superimposition is the ENVRI-RM+K model. This section summarizes the +K extension. It is presented in greater detail in Paper **V**.

3.2.1 Extension Subsystems

The +K extension introduces four subsystems: knowledge acquisition, knowledge curation, knowledge access, and knowledge processing. Unless stated otherwise, in this dissertation it is *situational* knowledge that is acquired, curated, accessed, and processed. In contrast to Paper **V**, the description here thus directly adopts the more narrow concept of situational knowledge.

The knowledge acquisition subsystem acquires situational knowledge from data. Acquiring situational knowledge is a process and consists of three sub processes: information attainment, information mapping, and knowledge composition. Information is attained from data, is mapped to atomic entities of a conceptual model, and is composed to structured entities of a conceptual model, i.e. situation knowledge objects or simply situations.

Similarly to the ENVRI-RM data curation subsystem, the knowledge curation subsystem facilitates quality control and preservation of situational knowledge. It also handles the representation of situational knowledge. The subsystem is however not limited to managing situational knowledge. It also manages domain knowledge, such as infon relations or domain rules, and foundational knowledge, such as a the fact that sensing devices are sensors and thus physical objects.

The knowledge access subsystem is concerned with the presentation and delivery of knowledge products, in particular situational knowledge products. Retrieval of knowledge is enabled by query and search tools. Such tools also support human or software agents in knowledge discovery, enabled by inspecting domain knowledge or by following semantic relations. For instance, domain

Figure 3.3: The functionality of the +K extension, and functionality partitioning amongst the four +K subsystems for the acquisition, curation, access, and processing of (situational) knowledge.

knowledge can be inspected for infon relations; infons with a certain relation can thus be discovered. Infons relate to objects, which may be situations. Thus, by following semantic relations it is possible to discover situations that are related, e.g. a situation in which an agent refers to another situation.

Classical reasoning (inference) services are part of the knowledge processing subsystem and include conceptual and rule-based reasoning. However, the subsystem is not limited to such type of knowledge processing. It can include spatio-temporal reasoning and specialized services for statistical analysis or visualization of situational knowledge. For instance, a knowledge processing subsystem may provide a service that computes how frequently people drive through storms. In addition, the knowledge processing subsystem can provide specialized services for situation reasoning.

Extension Functionality

Each of the four +K subsystems for the acquisition, curation, access, and processing of situational knowledge addresses a range of concerns and implements

specific functionality. Figure 3.3 visualizes how the functionality of the extension is partitioned amongst the four +K subsystems.

The knowledge acquisition subsystem is primarily concerned with attaining information from data, mapping attained information to atomic entities of a conceptual model, and composing mapped information to structured entities of a conceptual model, i.e. situations. For instance, the label returned in machine learning classification is a result of information attainment. A relevant individual, object in an infon, is a result of information mapping. An individual instance of the class STO `Situation` with relations to infons is a result of knowledge composition and is a situational knowledge object.

The primary concerns of the knowledge curation subsystem are knowledge quality checking, knowledge storage and preservation, knowledge representation, and knowledge identification. Knowledge quality checking may be implemented by software agents, human agents, or collaboratively. Knowledge storage and preservation is supported by a knowledge store, most obviously a knowledge base or an RDF database. Knowledge representation is conforming with the ontological framework centred around the STO. Knowledge identification is enabled by Internationalized Resource Identifiers (Dürst and Suignard, 2005, IRI), used to identify RDF resources. Situational knowledge is thus globally identified.

The primary concerns of the knowledge access subsystem are knowledge discovery and retrieval as well as knowledge publication. Functionality for knowledge discovery and retrieval operates on, and retrieves situational knowledge from, knowledge resources. The knowledge resource is, generally, the system associated with the knowledge storage and preservation functionality, and can be a distributed system of knowledge stores. The knowledge publication functionality addresses the publication of situational knowledge according to publication policies. A publication policy may specify situational knowledge to be publicly accessible or restricted. Situational knowledge may be published for download, typically over the web. Access may also be supported via (web) Application Programming Interface (API).

The knowledge processing subsystem is concerned with various forms of knowledge processing, in particular knowledge visualization and analysis and different forms of reasoning. Situational knowledge is generally located in time and space. Knowledge can thus be visualized along these two dimensions. Time lines and maps can support such visualization. Knowledge analysis depends on the domain and problem, which thus define the particular methods of interest in knowledge analysis. For instance, given situations involving classified vehicles travelling on a road section, and situational knowledge with information for vehicle speed, statistical analysis can be used to compute summary statistics, such as mean vehicle speed and standard deviation. Given a set $\mathcal{S}$ of situations, situation reasoning can support the inference of situations implied by $\mathcal{S}$. For

instance, given a set $\mathcal{S}$ of situations with information for the spatio-temporal location of storms and situations with information for the spatio-temporal location of drivers, situation reasoning can infer situations in which drivers are at higher risk, implied by situations in which storms and drivers overlap in space-time.

Ontology reasoning tasks are those typically supported by standard RDFS and OWL reasoners, such as Pellet (Sirin et al., 2007). In practice, implementations utilize ontology reasoning supported (if any) by the system associated with the knowledge storage and preservation functionality. The knowledge curation subsystem may materialize entailed knowledge. For instance, in addition to stating that a situation is an entity of a concept defined by STO, knowledge representation may also explicitly state that the situation is an entity of a concept defined by PROV-O. Similarly to ontology reasoning, rule-based reasoning is typically supported by RDFS and OWL reasoners. In practice, it is largely the system associated with the knowledge storage and preservation functionality that determines rule-based reasoning support. However, in specific applications rule-based reasoning can also be implemented in SPARQL or with domain program logic.

3.2.2 Extension Viewpoints

The +K extension specifies knowledge-based environmental research infrastructure from the science, information, and computational viewpoints.

Science Viewpoint

The science viewpoint intends to capture the requirements for the +K extension from the perspective of people, in particular researchers and citizens more generally. The extension defines five communities: knowledge acquisition, knowledge curation, knowledge publication, knowledge service provision, and knowledge usage. Each community is described for its roles and behaviours.

The *knowledge acquisition community* is who attains information from data, maps attained information to atomic entities of a conceptual model, and composes mapped information to structured entities of a conceptual model. Key *roles* include the attainer, mapper, and composer. Information is attained from data by an attainer, which is an active or a passive role. Of primary interest here is the passive role of attainer, i.e. the extractor. The mapper and the composer are generally passive roles. Key *behaviours* include knowledge acquisition, conceptual model extension, and software extension. Knowledge acquisition consists of three behaviours: information attainment, information mapping, and knowledge composition. These behaviours are performed by the three roles attainer, mapper, and composer, respectively. Conceptual model extension and software extension are behaviours performed by computer experts. An example of conceptual model extension is the instantiation of infon relations that are relevant

to situations in a knowledge acquisition problem. Software extension includes the implementation of software agents required for knowledge acquisition.

The *knowledge curation community* is who curates, maintains and archives knowledge. Key *roles* include the knowledge curator, knowledge representer, knowledge identifier, and the knowledge store. Of primary concern to the knowledge curator is the verification of situational knowledge resulting in the knowledge acquisition subsystem. In addition, the knowledge curator maintains domain knowledge managed by the knowledge curation subsystem. The knowledge representer is a passive role. Unless stated otherwise, the knowledge curation community represents situational knowledge according to the ontological framework centred around the STO. Knowledge identification occurs by means of IRI. It is software agents that create and assign IRIs to situational knowledge. The knowledge store is most obviously implemented by a knowledge base or an RDF database. However, the extension is not restricted to such knowledge store types. Key *behaviours* include knowledge quality checking, knowledge representation, knowledge identification, knowledge persistence, knowledge preservation. Quality checking is typically performed by an active role. However, as for data quality checking, software agents may quality check situational knowledge to some degree. The required degree of quality control largely depends on the performance of knowledge acquisition. The confidence in quality situational knowledge increases with greater knowledge acquisition performance. Knowledge representation, identification, and persistence are behaviours of distinct roles. In practice, these behaviours may be performed by one or more agents, in particular software agents. In addition to domain and situational knowledge, knowledge preservation is also concerned with the preservation of provenance information, and thus the agents and methods involved in knowledge acquisition and processing.

The *knowledge publication community* is who assists knowledge publication, discovery and access. Key *roles* include the knowledge publication repository and the knowledge consumer. Situational knowledge can be published in various forms. Publishing situational knowledge as RDF files for download is perhaps the most straightforward form. In alternative to RDF files for download, situational knowledge may also be published via a SPARQL endpoint. Knowledge consumers are either active or passive roles that receive and use situational knowledge published by the knowledge publication repository. Key *behaviours* include knowledge publication and knowledge discovery and access.

The *knowledge service provision community* is who provides various services, applications and software tools used to process knowledge. Key *roles* include the knowledge provider and the software engineer. Situational knowledge processing, e.g. visualization or analysis, can require domain program logic; hence the role of software engineers. Software implementation is a *behaviour* of the knowledge service provision community.

The *knowledge usage community* is who makes use of knowledge and service products, and transfers knowledge into understanding. The roles and behaviours are same as those of the ENVRI-RM data usage community.

Information Viewpoint

The information viewpoint intends to provide an abstract model for the shared information objects that are relevant to the +K extension by specifying their types and relations between types. The information viewpoint discusses the following aspects of the extension: components, dynamic schemata, static schemata, subsystem schemata. This section summarizes the components aspect. Paper **V** discusses the other aspects as well.

The components aspect of the information viewpoint organises the model elements that are relevant to the extension into four groups: information objects, information action types, information object instances, and knowledge states.

Information objects are defined to capture three types of information relevant to the extension. The first type of information includes specifications for knowledge acquisition, knowledge curation, knowledge access, and knowledge processing. Specifications are documents, created by experts, and describe the objects and methods involved in behaviours, e.g. knowledge acquisition. The second type of information captured by information objects includes the types of data, information, and knowledge objects, specifically attained information, mapped information, and composed knowledge. Attained information objects are the result of information attainment, which executes on data objects. Attained information objects are generally values of some primitive data type. For instance, the label returned in machine learning classification is an attained information object. Attained information objects are mapped to atomic entities of a conceptual model. The result are mapped information objects. For instance, an attained information object can be mapped to a relevant individual, object in an infon of a situation. Finally, mapped information objects are composed to structured entities of a conceptual model, i.e. situations. Situational knowledge consists of information attained by one or more extractors, and mapped to one or more atomic entities of a conceptual model. Situational knowledge is thus a composition of information objects. It is the input and output to knowledge curation, access, and processing. In the information viewpoint, data objects, attained and mapped information objects, and knowledge objects (situations) are specializations of information objects. We may thus speak of information objects without further qualifying the specific type. The third type of information captured by information objects includes information for knowledge provenance used to record state changes of information objects, in particular the state changes of attained information objects, mapped information objects, and composed knowledge objects.

Information action types model how data, information, and knowledge are processed in the system. The most fundamental information action type is to perform knowledge acquisition. Additional important information action types are for representing, managing, and processing situational knowledge. Knowledge representation involves one or more knowledge representation languages, related technologies, and the ontological framework centred around the STO. Management includes, in particular, storing, checking the quality, and querying situational knowledge. Information objects may exist as multiple *instances*. One purpose of instances is to record *knowledge state* changes as effects of actions. For instance, a knowledge object resulting in knowledge acquisition is in state acquired. As effect of the process knowledge action the knowledge object is in state processed.

Computational Viewpoint

The computational viewpoint describes the computational objects of the +K extension, and computational object interfaces.

The knowledge acquisition subsystem provides functionality for attaining information from data, mapping attained information to atomic entities of a conceptual model, and composing mapped information to structured entities of a conceptual model. Computationally, knowledge acquisition is described as sets of information attainers, information mappers, and knowledge composers associated with knowledge acquisition controllers. A knowledge acquisition controller receives and directs data to attainers, attained information to mappers, and mapped information to composers. A knowledge acquisition controller returns composed knowledge. A knowledge attainer receives data and attains information. Its output are attained information objects. Generally, information is attained by means of computational models, e.g. data-driven or physically-based models. However, a human agent may also attain information. An information mapper receives attained information and maps information. Its output are mapped information objects. A knowledge composer receives mapped information and composes knowledge. Its output are composed knowledge objects, i.e. situations.

The knowledge curation subsystem provides functionality to persist and preserve situational knowledge. Computationally, knowledge curation is handled by a set of knowledge store controllers monitored and managed by a set of knowledge curation services, specifically knowledge annotation services and knowledge transfer services. A knowledge annotation service implements the functionality required to annotate situational knowledge. The service is primarily intended for use in knowledge quality checking to update inconsistent or inaccurate situational knowledge. A knowledge transfer service supports the registration, deregistration, and execution of knowledge transporters, such as knowledge collectors, importers, and exporters.

The knowledge access subsystem provides knowledge brokers that act as intermediaries for access to situational knowledge managed by the knowledge curation subsystem. Knowledge brokers intercede between the knowledge access subsystem and the knowledge curation subsystem. They implement the functionality required to negotiate data transfer and requests directed at knowledge curation services on behalf of agents.

The knowledge processing subsystem is computationally described as a set of knowledge processing controllers monitored and managed by a knowledge processing coordination service. The coordination service delegates processing tasks obtained by the knowledge processing subsystem to particular execution resources. A knowledge processing controller "encapsulates the functions required for using an execution resource." An execution resource is "any computing platform that can host some process."

3.3 FRAMEWORK IMPLEMENTATION

The Wavellite software framework was developed to support the implementation of environmental monitoring software systems that aim at the acquisition of situational knowledge from environmental sensor network data, as well as the curation, access, and processing of situational knowledge.

ENVRI-RM+K is used to present the Wavellite software architecture and implementation. Wavellite borrows elements of both ENVRI-RM and the +K extension. This section follows the structure introduced by ENVRI-RM, and adopted by the +K extension, to discuss Wavellite subsystems, and Wavellite from science, information, and computational viewpoints.

It is important to underscore that Wavellite is not an environmental research infrastructure and, specifically, it is not an implementation of ENVRI-RM or the +K extension. Wavellite was designed to support *core* functionality required in systems for situation awareness in environmental monitoring. Much of the functionality ENVRI-RM defines as the minimal model for an environmental research infrastructure is indeed not supported by Wavellite. Moreover, in contrast to ENVRI-RM, Wavellite is not built on service-oriented principles. Similar to ENVRI-RM, Wavellite supports the acquisition of data collected from sensor networks as well as the curation, access, and processing of data. However, such functionality is primarily intended to support knowledge acquisition whereas ENVRI-RM emphasises different data processing functionality, such as data visualization. Furthermore, Wavellite is tailored for the acquisition of a particular kind of knowledge: situational knowledge. In contrast, an environmental research infrastructure may want to support the acquisition of other kinds of knowledge. Still, for the presentation of the Wavellite software architecture, ENVRI-RM and the +K extension are useful reference models because of considerable shared functionality, the explicit inclusion of sensors and the science

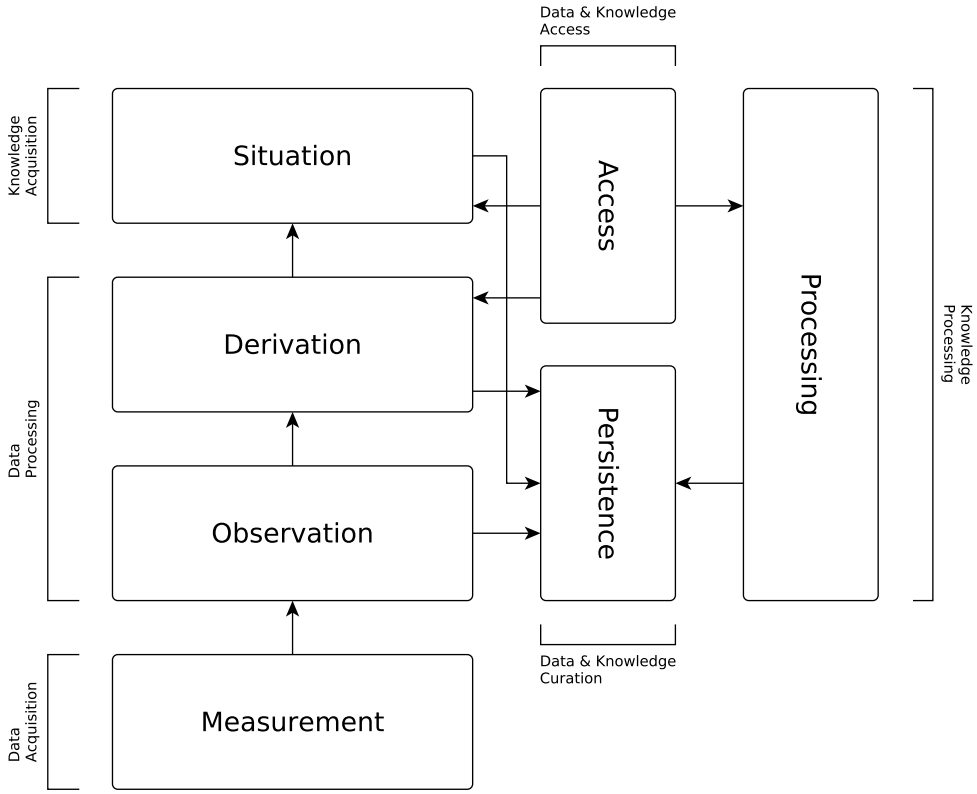


Figure 3.4: The Wavellite layers, their structure and mapping to ENVRI-RM+K subsystems. The layers of measurement, observation, derivation, and situation build on each other and are responsible for the acquisition of data from environmental sensor networks, the processing of data, and the acquisition of situational knowledge. The persistence and access layers support the storage and retrieval of data and knowledge. The processing layer is responsible for situational knowledge processing and builds on the persistence and access layers.

community in ENVRI-RM, and because it enables an alignment of Wavellite with environmental research infrastructure.

The Wavellite software framework is structured in seven layers: measurement, observation, derivation, situation, persistence, access, and processing. The four layers of measurement, observation, derivation, and situation build on each other, from measurement to situation. The persistence and access layers are vertical and serve the other five layers. The processing layer builds on the access and persistence layers. Layers consist of components. Components execute and emit data, information, and knowledge objects. Components make use of modules. Modules implement functionality.

3.3.1 Implementation Subsystems

Wavellite layers can be mapped to ENVRI-RM+K subsystems, i.e. data *and* knowledge acquisition, curation, access, and processing. Figure 3.4 is a graphical overview of the seven Wavellite layers, their structure and interaction, and their mapping to ENVRI-RM+K subsystems.

The *measurement layer* maps to the data acquisition subsystem. It is concerned with the data collection and data transmission functionality of the data acquisition subsystem. Data collection is a "functionality that obtains digital values from a sensor instrument, associating consistent timestamps and necessary metadata." Data transmission is a "functionality that transfers data over a communication channel using specified network protocols." The measurement layer acts as interface between sensor networks and Wavellite. It abstracts from the heterogeneity of sensors, communication channels and network protocols, and data encoding. It ensures that collected data are translated into measurement results. Measurement results are data objects and are communicated to the observation layer.

The *observation layer* and *derivation layer* map to the data processing subsystem. They are concerned with data processing, in particular the data analysis and scientific workflow enactment functionality of the data processing subsystem. Data analysis is a "functionality that inspects, cleans, transforms data." Scientific workflow enactment "supports [the] composition and execution [of] a series of computational or data manipulation steps." The observation layer abstracts from the heterogeneity of sensor data by mapping measurement results to sensor observations. The term `SensorObservation` is aligned with the term `Observation` of the SSN ontology. Sensor observations are data objects and are communicated to the derivation and persistence layers.

The derivation layer maps sensor observations to dataset observations. It abstracts the sensor aspect of sensor data. The term `DatasetObservation` is aligned with the term `Observation` of the QB vocabulary. In addition, the derivation layer supports the composition and execution of data manipulation steps—for inspection, cleaning, and transformation of data, in particular. This feature of the derivation layer is implemented as series of derivation steps, whereby at each step an input set of dataset observations is processed to an output set of dataset observations. Dataset observations are data objects and are communicated to components of the derivation layer as well as to the situation and persistence layers.

The *situation layer* maps to the knowledge acquisition subsystem. It is concerned with the information attainment, information mapping, and knowledge composition functionality of the knowledge acquisition subsystem. Information is attained from dataset observations and is composed to situational knowledge, i.e. individual situations instances of STO `Situation`. Situations are knowledge objects and are communicated to the persistence layer.

The *persistence layer* maps to the data and knowledge curation subsystems. It is primarily concerned with identification and storage of data and knowledge. Identification is a functionality that assigns global unique identifiers to information objects. Storage and preservation is a functionality that persists information objects and provides access to them upon request.

The *access layer* maps to the data and knowledge access subsystems. It is primarily concerned with access control as well as publication and discovery of, and access to, data and knowledge. Access control is a "functionality that approves or disapproves of access requests based on specified access policies." Publication is a "functionality that provides clean, well-annotated, anonymity-preserving [data and knowledge sets] in a suitable format." Discovery and access is a "functionality that retrieves requested [data or knowledge] from a [data and knowledge] resource by using suitable search technology."

The *processing layer* maps to the knowledge processing subsystem. It is concerned with processing situational knowledge. The layer includes generic and application-specific software agents that process situational knowledge. For example, a generic tool that visualizes situational knowledge in space-time is a software agent of the processing layer.

3.3.2 Implementation Viewpoints

This section presents Wavellite from the three ENVRI-RM+K viewpoints, namely the science, information, and computational viewpoints. Being a software framework, of particular interest are the information and computational viewpoints. However, the section first presents the science viewpoint and discusses the communities, roles, and behaviours that are relevant to Wavellite applications.

Science Viewpoint

In Wavellite applications, all ENVRI-RM+K communities are relevant, albeit to a varying extent. Given that the primary purpose of Wavellite applications is situational knowledge acquisition and processing, the most important communities are those involved in the preparation and execution of tasks related to these processes.

The most relevant roles in the *data acquisition community* are the sensor, sensor network, and the measurement layer. A sensor "measures a physical quantity and converts it into a signal which can be read by an observer or by an (electronic) instrument." A sensor network "is a network consisting of distributed autonomous sensors to monitor physical or environmental conditions." In Wavellite applications, sensor and sensor network are both passive roles. The measurement layer collects sensor data transmitted over a communication channel, and is a passive role in the data acquisition community. In Wavellite applications, the

communication between the measurement layer and sensors is, generally, direct. It is components of the measurement layer that connect to sensors and collect data.

Other roles in the data acquisition community are relevant, albeit to a lesser extent. The technician, a "person who develops and deploys sensor instruments [...]," is relevant to Wavellite applications. However, in general sensor instruments are deployed prior to application development, and from the perspective of the application their set-up is determined. In addition to technicians, the environmental scientist is an important role of the data acquisition community for Wavellite applications because such persons hold domain knowledge, such as knowledge about the properties observed by sensors or about the environmental phenomena involved in situations.

Among the behaviours of the data acquisition community, the most important is data collection. ENVRI-RM states that data collection is a "behaviour performed by a data collector that obtains digital values from a sensor instrument [or a human sensor], associating consistent timestamps and necessary metadata." Data collector is an active role. In Wavellite applications, data collection is generally performed by a passive role—a software agent and, specifically, a component of the measurement layer.

The most relevant roles in the *data curation community* are the data curator and the persistence layer. The data curator—an "active role, which is a person who verifies the quality of the data, preserves and maintains the data as a resource, and prepares various required data products"—is an important role because it is critical to verify that data processing computations required for knowledge acquisition are implemented correctly. The persistence layer stores, manages and ensures access to information objects produced in Wavellite applications. It is a passive role in the data curation community.

The storage—a "passive role, which includes memory, components, devices and media that retain digital computer data used for computing for some interval of time"—is a relevant role as well. In practical applications, the storage is typically provided by a workstation or server, perhaps the one that executes the application. Similarly, the storage administrator—an "active role, which is a person who has the responsibilities to design data storage, tune queries, perform backup and recovery operations [...]"—is a relevant role but in practical applications arguably not as relevant as the persistence layer.

Among the behaviours of the data curation community, the most important are data quality checking and data preservation. According to ENVRI-RM, both behaviours are performed by a data curator, which is an active role. Data quality checking "detects and corrects (or removes) corrupt, inconsistent or inaccurate records from data sets." Data preservation "deposits (over long-term) the data and metadata or other supplementary data and methods according to specified policies, and makes them accessible on request." In Wavellite applications, data

are deposited by software agents, and data preservation is thus more accurately described as a behaviour of a passive role—a component of the persistence layer.

Because the primary aim of Wavellite applications is knowledge acquisition, and its curation, access, and processing, data publication is not a primary concern. Therefore, the *data publication community* holds a relatively minor role in Wavellite applications. The access layer is the most important role of the data publication community. It is a passive role and enables the discovery and retrieval of (scientific) data. A second role with some importance in concrete applications is the data consumer. A data consumer receives and uses data. The most relevant behaviour of the data publication community is data discovery and access, which "retrieves requested data from a data resource by using suitable search technology."

The *data service provision community* is critical in environmental research infrastructure as it "provides various services, applications and software/tools to link and recombine data and information in order to derive knowledge" for a wide range of services, including data assimilation, analysis, mining, and extraction. Wavellite applications also rely on software agents that 'recombine' data. However, the type of software agents is constrained to meet the purpose of Wavellite applications, i.e. situational knowledge acquisition. Data processing occurs within the observation layer and, in particular, the derivation layer. These two layers are passive roles of the data service provision community. The most important behaviours of the data service provision community are the translation of measurement results into sensor observations, performed within the observation layer; the translation of sensor observations into dataset observations, performed within the derivation layer; and the derivation of input sets of dataset observations into output sets of dataset observations, performed within the derivation layer.

Akin to the data publication community, the *data usage community* holds a relatively minor role in Wavellite applications. This is because it is situational knowledge, rather than data, that is of primary interest. Nevertheless, a relevant role is the technologist or engineer—an "active role, which is a person who develops and maintains the research infrastructure." In Wavellite applications, the technologist is responsible for the implementation and maintenance of the application, and is thus a primary user of data. Other roles are potentially relevant, in particular the scientist or researcher.

The passive roles involved in knowledge acquisition, namely the extractor, mapper, and composer are key roles in the *knowledge acquisition community*. These are software agents that operate within the situation layer. The situation layer is an additional important passive role of the knowledge acquisition community. In Wavellite applications, extractors, mappers, and composers are responsible for the acquisition of situational knowledge from dataset observations. The result are situations. Relevant active roles in the knowledge acqui-

sition community include computer and domain experts. Experts design and implement software agents for knowledge acquisition. Thus, experts are particularly important in the design and implementation phases of Wavellite applications. During runtime, it is primarily passive roles that perform knowledge acquisition, which thus operate autonomously.

Being the core goal of the Wavellite software framework, knowledge acquisition is arguably the most important behaviour of the knowledge acquisition community. It consists of the information attainment, information mapping, and knowledge composition behaviours. Such behaviours are performed by attainers, mappers, and composers, respectively. However, applications generally need to extend the Wavellite software framework with domain program logic to implement specific knowledge acquisition tasks. Software extension, performed by computer experts, is thus an important behaviour.

A role of particular interest in the *knowledge curation community* is the knowledge representer. It is a passive role, a software agent of the persistence layer that represents sensor observations, dataset observations, and situations according to the syntax and semantics of data models, knowledge representation languages, and ontologies. In Wavellite applications, RDF is the data model, RDFS and OWL are the knowledge representation languages, and the SSN ontology, QB vocabulary, STO, OWL-Time, GeoSPARQL, and PROV-O are the ontologies. In addition to being a passive role of the data curation community, the persistence layer is also a passive role of the knowledge curation community.

Knowledge identification is a behaviour of the knowledge curation community by which situational knowledge is identified. By building on RDF, knowledge identification in Wavellite is by means of IRI. Therefore, situations are resources identified by IRI. Infons, relations, objects, attributes, and values related to situations are also resources identified by IRI. Knowledge persistence is another relevant behaviour of the knowledge curation community, performed by knowledge stores. In Wavellite applications, the knowledge store is generally a knowledge base (Baader et al., 2007) and is a software agent of the persistence layer. At a minimum, the knowledge store must support the persistence and retrieval of RDF.

The key role in the *knowledge publication community* is the access layer. It is a passive role and enables the retrieval and discovery of knowledge. In Wavellite applications, retrieval and discovery is generally by means of SPARQL. As for data, the publication of knowledge is not of primary concern in Wavellite applications, especially those discussed in Chapter 4. Thus, roles such as the knowledge publication repository or the knowledge consumer, as well as the knowledge publication behaviour, are not particularly relevant.

The knowledge provider is an important role of the *knowledge service provision community*. In Wavellite applications, the knowledge provider is generally a passive role, a software agent that provides situational knowledge to service

providers. The service provider is an active or passive role in the ENVRI-RM data service provision community, "an entity providing the services to be used." In Wavellite applications, service providers are generally passive roles and entities of the processing layer. For instance, a service that visualizes situational knowledge is a service provider of the processing layer. The processing layer itself is a passive role of the knowledge service provision community. The software engineer is a further role of the knowledge service provision community, relevant to Wavellite applications. It is an active role, a person who implements domain specific services, applications, and software tools, such as for situational knowledge visualization, analysis, and reasoning.

Roles in the *knowledge usage community* are equal to those of the ENVRI-RM data usage community. All roles are potentially interesting, including educators, decision makers, consultants, and the general public. However, of most relevance to Wavellite applications are the technologist or engineer and the scientist or researcher, for instance aerosol scientists and agricultural advisers.

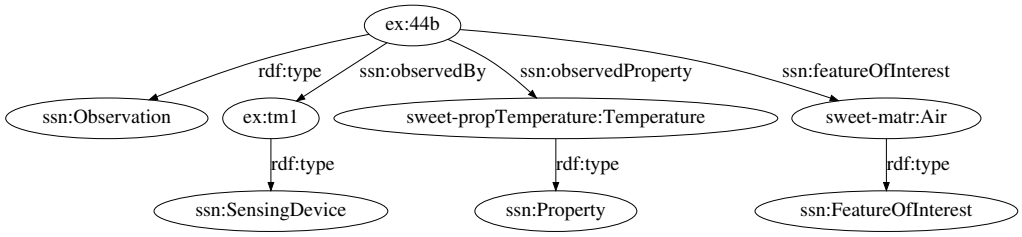
Information Viewpoint

The Wavellite information viewpoint is discussed for the aspects introduced by ENVRI-RM, namely components, dynamic schemata, static schemata, and subsystem schemata.

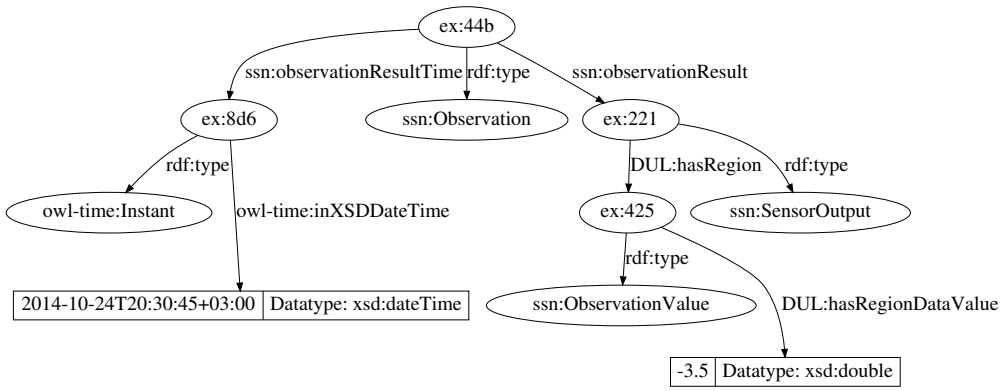
Components Wavellite introduces a number of *information objects*. They can be grouped into the three ENVRI-RM categories, which are extended with knowledge: the meta information of data and knowledge collections, the data and knowledge processed by the system, and the information used for the management of data and knowledge. The most important category is the data and knowledge processed by the system.

Information objects of the first category, i.e. meta information of data and knowledge collections, include specifications for data processing and for knowledge acquisition, as well as the ENVRI-RM specification of observation and description of the measurement procedure. An important information object of this category is the `DataStructureDefinition`, a term of the QB vocabulary. A data structure definition is meta information for one or more data collections, called datasets. Information objects of interest of the third category, i.e. information used for the management of data and knowledge, are data and knowledge provenance objects. Instances of data and knowledge provenance objects are individuals of PROV-O `Entity`.

Information objects of the second category, i.e. data and knowledge processed by the system, are of primary interest here and are discussed in details.



(a) The sensing device, property, and feature of interest related to the sensor observation.



(b) The temporal location and observation value related to the sensor observation.

Figure 3.5: A represented sensor observation for air temperature observed at a particular point in time. For better readability, the sensor observation is split into two graphs. The graphs can be joined via node `ex:44b`.

The pair

$$\mathrm{M}_r = (v_m, c(v_m)) \tag{3.1}$$

is a `MeasurementResult` and is a data object. It consists of a measurement value, v_m, and the context of v_m, $c(v_m)$. A measurement value is a number assigned in measurement (Finkelstein, 1982). It is generally a number of primitive type `double`. The tuple $c(v_m) = (s, p, f, l_t, l_s, q)$ consists of objects for a sensor, s, a property, p, a feature, f, a temporal location, l_t, a spatial location, l_s, and a quality, q.

The `SensorObservation` is a data object with semantics aligned with the term SSN `Observation`. Formally it is the tuple

$$\mathrm{O}_s = (s_o, s, p, f, l_t, l_s, q) \tag{3.2}$$

consisting of a sensor output, s_o, and a sensor, a property, a feature, a temporal location, a spatial location, and a quality. A sensor observation relates to the sensor that made the observation, the observed property, the monitored feature,

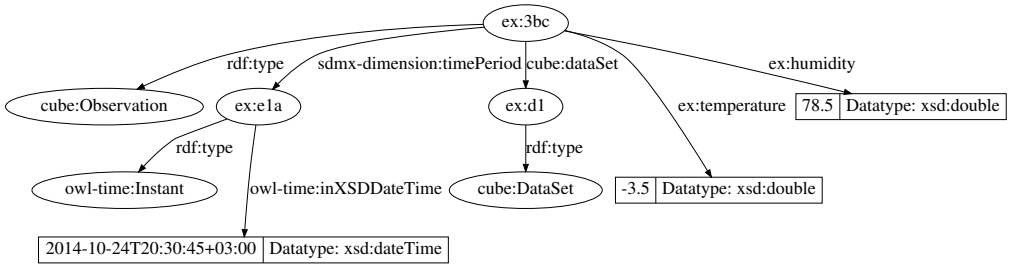


Figure 3.6: A represented dataset observation of dataset `ex:d1` with component property values for temporal location, temperature, and humidity.

the sensor output and observation value, the time and place where the observation was made, and the quality of observation. Figure 3.5 is an example of a *represented* sensor observation for air temperature observed at a particular point in time. The sensor observation is represented in RDF according to the SSN ontology. The sensing device, namely the thermometer `ex:tm1`, is considered to be at a fixed point in space. The spatial location is thus preferably modelled as metadata of the sensing device, rather than of the sensor observation.

The `DatasetObservation` is a data object with semantics aligned with the term QB `Observation`. It is the object from which knowledge is attained. Formally it is the pair

$$\mathrm{O}_d = (d, \mathcal{C}) \tag{3.3}$$

consisting of a dataset, d, and a set, $\mathcal{C}$, of dataset observation components. A dataset observation component, $c \in \mathcal{C}$, is a pair $c = (c_p, p_v)$ consisting of a component property, c_p, and a component property value, p_v. The component properties of a dataset observation must be distinct. Considered are two component property types, namely *dimension* component property, c_p^d, and *measure* component property, c_p^m. Figure 3.6 is an example of a *represented* dataset observation. It relates to dataset `ex:d1` and holds a set of component property values for temporal location, temperature, and humidity. The dataset observation is represented in RDF according to the QB vocabulary.

Attained information objects are of some primitive data type, such as `String` or `Double`. For instance, the result of classifying a `DatasetObservation` using, e.g., a trained MLP artificial neural network is a label for the class, and the label is of primitive data type `String`. Mapped information objects are instances of STO `Object`. Concrete examples include `RelevantIndividual`, `Attribute`, `Value`, or `Situation`. STO objects may be OWL-Time or GeoSPARQL entities.

`Situation` is a knowledge object with semantics aligned with the term STO `Situation`. In Wavellite, `Situation` is a composed knowledge object. A situation, S, is said to support a set of infons $\mathcal{I}$. An infon $i \in \mathcal{I}$ is a tuple $i = (r, t, \mathcal{O})$

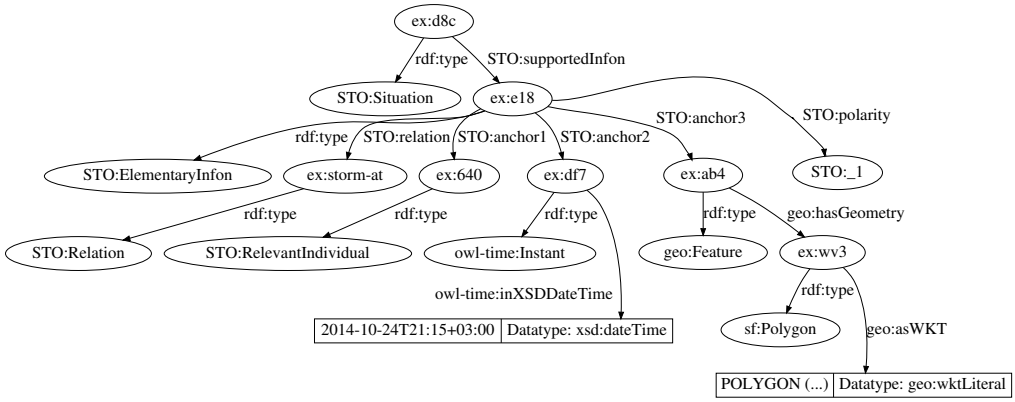


Figure 3.7: A represented situation for a storm at a particular point in time and space. For better readability, the polygon coordinates are omitted.

consisting of a relation, r, a polarity, t, and a set of relevant objects, $\mathcal{O}$. A relevant object, $o \in \mathcal{O}$, is any object that is relevant to the infon i, e.g. an object representing a particular manifestation of an environmental phenomenon, a temporal location, a spatial location, a sensor. Relevant objects $\mathcal{O}$ stand in relation r if $t = 1$ and do not stand in r if $t = 0$.

Figure 3.7 is an example of a *represented* situation for a storm at a particular point in time and space. The situation supports an infon with `storm-at`-relation and three objects: a relevant individual representing the storm (`ex:640`), a temporal location, and a spatial location. The relevant individual can be annotated with attributes, such as the area of the storm. The situation is represented in RDF according to the STO.

Wavellite introduces a number of *information action types*. There exist three fundamental types: translation, processing, and acquisition. The translation of measurement result M_r into sensor observation O_s

$$\mathcal{T}_m : \mathrm{M}_r \mapsto \mathrm{O}_s \tag{3.4}$$

is a mapping of elements of the pair $(v_m, c(v_m))$ to elements of the sensor observation O_s, specifically $v_m \mapsto s_o$ and the mapping of the elements s, p, f, l_t, l_s, q onto themselves. The translator $\mathcal{T}_m$ is used at the observation layer.

The translation of sensor observation O_s into dataset observation O_d

$$\mathcal{T}_o(d, \mathcal{C}_p) : \mathrm{O}_s \mapsto \mathrm{O}_d \tag{3.5}$$

is a mapping of elements of the sensor observation O_s to components $c \in \mathcal{C}$ of the dataset observation $\mathrm{O}_d = (d, \mathcal{C})$, element of dataset d. The translator $\mathcal{T}_o(d, \mathcal{C}_p)$ accepts two parameters, namely a dataset d and a set $\mathcal{C}_p$ of component properties. For sensor observations with constant spatial location, the O_s elements l_t and s_o

are mapped to a dimension component property, c_p^d, and a measure component property, c_p^m, respectively, whereby $c_p^d, c_p^m \in \mathcal{C}_p$. In this case, dataset observations form a time series that can be graphed with c_p^d as domain (x-axis) and c_p^m as range (y-axis). For sensor observations with variable spatial location, the O_s entity l_p is mapped to an additional dimension component property. The translator $\mathcal{T}_o$ is used at the derivation layer.

The processing of a source set of dataset observations ($'$) into a target set of dataset observations ($''$), elements of dataset d

$$\mathcal{P}_o(d) : \{\mathrm{O}_d^i\}' \mapsto \{\mathrm{O}_d^j\}'' \tag{3.6}$$

whereby $i = 1, \ldots, p$ and $j = 1, \ldots, q$, is a formalization of the ENVRI-RM process data information action type. It is used at the derivation layer to process dataset observations.

Several concrete $\mathcal{P}_o(d)$ implementations are of interest. $\texttt{Merge}(d) : \{\mathrm{O}_d^i\}' \mapsto \{\mathrm{O}_d^1\}''$ processes a set of dataset observations ($'$) into a singleton ($''$). The singleton consists of a dataset observation $\mathrm{O}_d = (d, \mathcal{C})$ element of dataset d with set $\mathcal{C}$ for the union of components $c = (c_p, p_v)$ of dataset observations $\{\mathrm{O}_d^i\}'$. It is assumed that component properties common to multiple dataset observations in $\{\mathrm{O}_d^i\}'$ have the same component property value, p_v. Otherwise, the singleton $\{\mathrm{O}_d^1\}''$ will non-deterministically include one component, c, with component property common to multiple dataset observations in $\{\mathrm{O}_d^i\}'$. For instance, if c_p is a dimension component property and the related p_v a temporal location then p_v should be same for all dataset observations in $\{\mathrm{O}_d^i\}'$ that have c_p in common. $\texttt{Aggregate}(d, c_p^d, \mathrm{F}, \mathrm{P}) : \{\mathrm{O}_d^i\}' \mapsto \{\mathrm{O}_d^j\}''$ applies, for each applicable component property c_p of components $c = (c_p, p_v)$ of dataset observations $\{\mathrm{O}_d^i\}'$ within time period P, the function F to the corresponding component property values p_v. It is assumed that the temporal location dimension component property c_p^d is common to all dataset observations in $\{\mathrm{O}_d^i\}'$. The resulting set $\{\mathrm{O}_d^j\}''$ consists of dataset observations with set $\mathcal{C}$ including the component with dimension component property c_p^d and related temporal location p_v rounded to the time period P as well as, for each applicable component property c_p, the related component property value p_v for the result of the function F. Examples for functions F include mean, max, or min. Examples of time periods P include minute, hour, or day. In addition to `Merge` and `Aggregate`, $\mathcal{P}_o(d) : \{\mathrm{O}_d^i\}' \mapsto \{\mathrm{O}_d^j\}''$ implementations may be for interpolation, filtering, Fourier transform, and other data processing techniques.

The acquisition of situational knowledge from a source set $\{\mathrm{O}_d^i\}$ of dataset observations into a target set $\{\mathrm{S}^j\}$ of situations

$$\mathcal{A}_s : \{\mathrm{O}_d^i\} \mapsto \{\mathrm{S}^j\} \tag{3.7}$$

whereby $i = 1, \ldots, p$ and $j = 1, \ldots, q$, is a formalization of the perform knowledge acquisition information action type, and is used at the situation layer. The $\mathcal{A}_s$ information action type performs the information attainment, information mapping, and knowledge composition information action types. Computational models of interest to $\mathcal{A}_s$ implementations include data-driven models, such as machine learning classification or complex event processing (Luckham, 2002, CEP), and physically-based models. For instance, $\texttt{ML}(\mathrm{C}, \mathrm{T}) : \{\mathrm{O}_d^i\} \mapsto \{\mathrm{S}^j\}$ is for machine learning and uses the classifier C and the training dataset T to classify dataset observations $\{\mathrm{O}_d^i\}$ and acquire situational knowledge $\{\mathrm{S}^j\}$. An example classifier C is the MLP artificial neural network.

Other ENVRI-RM+K information action types are relevant to Wavellite. For instance, the ENVRI-RM perform measurement observation information action type produces measurement results, and is used at the measurement layer; the store data and store knowledge information action types are used at the persistence layer; the query data and query knowledge information action types are used at the access layer.

In Wavellite applications, information objects exist as *information object instances*. They are input to and output of information action types. It is instances that are translated, processed, acquired, stored, queried, represented. Of particular interest within the framework are measurement result, sensor observation, dataset observation, and situation instances. These are objects, instances of corresponding classes of the Wavellite software framework. As a result of representation, sensor observations, dataset observations, and situations are individuals instances of the concepts SSN `Observation`, QB `Observation`, and STO `Situation`, respectively.

Various ENVRI-RM *data states* and *knowledge states* of the +K extension are relevant to Wavellite. Of particular interest are the data states raw, mapped, and processed and the knowledge states acquired and processed. Measurement results are in raw data state. Sensor observations are in mapped data state. Dataset observations are in processed data state. Situations are either in acquired or processed knowledge state, depending on whether a situation is the output of the perform knowledge acquisition or the process knowledge information action types, respectively.

Dynamic Schemata The design and deployment of sensor networks for measurement is generally completed before a Wavellite application is designed, implemented, and deployed. The sensor network is in operation, and it is known what data can be obtained via which communication channel. The Wavellite application is then developed in order to implement situational knowledge acquisition problems. The specifics of a Wavellite application depend on the addressed problem. However, at a minimum, a Wavellite application acquires and stores situational knowledge.

Situational knowledge acquisition tasks need to be first specified. The result is an information object, namely a specification for knowledge acquisition. It is a description of knowledge acquisition in the system and includes information about the Wavellite components as well as the information objects involved in information attainment, information mapping, and knowledge composition. Additionally, it must be specified how acquired situational knowledge is persisted. The result is an information object, namely a specification for knowledge curation. Given the specifications, the Wavellite application can be developed. The application needs to extend the Wavellite framework in order to implement domain knowledge and program logic according to the specifications. An implemented and tested Wavellite application can be executed.

A Wavellite application may *not* need to implement functionality of the measurement, observation, and derivation layers. This is the case when there exist dataset observations that do not need processing in order to serve knowledge acquisition. In other words, the Wavellite application is presented with dataset observations that serve as input to knowledge acquisition. In such applications, the situation layer performs knowledge acquisition, which entails performing information attainment, information mapping, and knowledge composition. Dataset observations serve as input to information attainment, which results in attained information objects. Attained informations objects serve as input to information mapping, which results in mapped informations objects. Mapped information objects serve as input to knowledge composition, which results in composed knowledge objects, i.e. situations. Situations are then represented and stored.

A Wavellite application may need to implement functionality of the measurement, observation, and derivation layers. It may only need to implement functionality of the derivation layer or it may also need to implement functionality of the observation and measurement layers. All layers are involved in applications that collect data from sensors. In such applications, the measurement layer performs data collection from sensors, associates consistent timestamps and necessary metadata to the obtained digital values, and returns measurement results. Measurement results are then translated into sensor observations by the observation layer. The system may represent and store sensor observations. Sensor observations are then translated into dataset observations by the derivation layer. The system may represent and store dataset observations. Generally, dataset observations need to be processed prior to knowledge acquisition. Such processing typically consists of several processing steps, as detailed by the specification for data processing information object. Dataset observation processing is executed by the derivation layer.

Beyond knowledge acquisition, a Wavellite application may need to process situational knowledge. In such applications, the processing layer uses the access layer to query situational knowledge and performs a process knowledge information action type. Examples for knowledge processing include visualiza-

tion, analysis, or reasoning. Wavellite applications may record the provenance of sensor observations, dataset observations, and situations during their life-cycle. The result of the track provenance information action type are data provenance or knowledge provenance information objects. Such objects can be represented, stored, queried, and potentially processed.

Static Schemata ENVRI-RM provides a set of constraints for data collection. The model underscores the importance of recording information about the measurement set-up, including information about involved sensing devices. Such information is important also in Wavellite applications. At the measurement layer, a measurement result must have a measurement value. While it arguably depends on the application, in general the context of the measurement value must be complete with information about the sensing device, the observed property and feature, and temporal location. Such information is thus associated with the measurement value. Spatial location must be included if the sensing device is attached to a mobile platform. Otherwise it is optional. Quality is optional.

At the observation layer, it is required that all obtained measurement results are translated into sensor observations. Measurement result elements must be correctly mapped to sensor observation elements. Sensor observations and their elements must have associated identifiers. At the derivation layer, it is required that all obtained sensor observations are translated into dataset observations. Sensor observation elements, such as temporal location and sensor output, must be correctly mapped to dataset observation elements, i.e. component property values. Resulting dataset observations must relate to a dataset. Dataset observations and their elements must have associated identifiers. At the situation layer, it is required that all attained information is appropriately mapped to atomic entities of a conceptual model and composed to situations. Situations and their elements must have associated identifiers.

At the persistence layer, sensor observations, dataset observations, and situations must be represented according to the SSN ontology, the QB vocabulary, and the STO, respectively. Temporal and spatial information associated with sensor observations, dataset observations, and situations must be represented according to OWL-Time and GeoSPARQL, respectively. Provenance information about sensor observations, dataset observations, and situations must be represented according to PROV-O. Representation is required to associate IRIs to resources so that resources are identified. All RDF statements resulting from the representation of sensor observations, dataset observations, and situations obtained by the persistence layer must be persisted by the data and knowledge stores.

Subsystem Schemata Subsystem schemata regroup Wavellite information objects and information action types into Wavellite layers. Discussed are the measurement, observation, derivation, and situation layers.

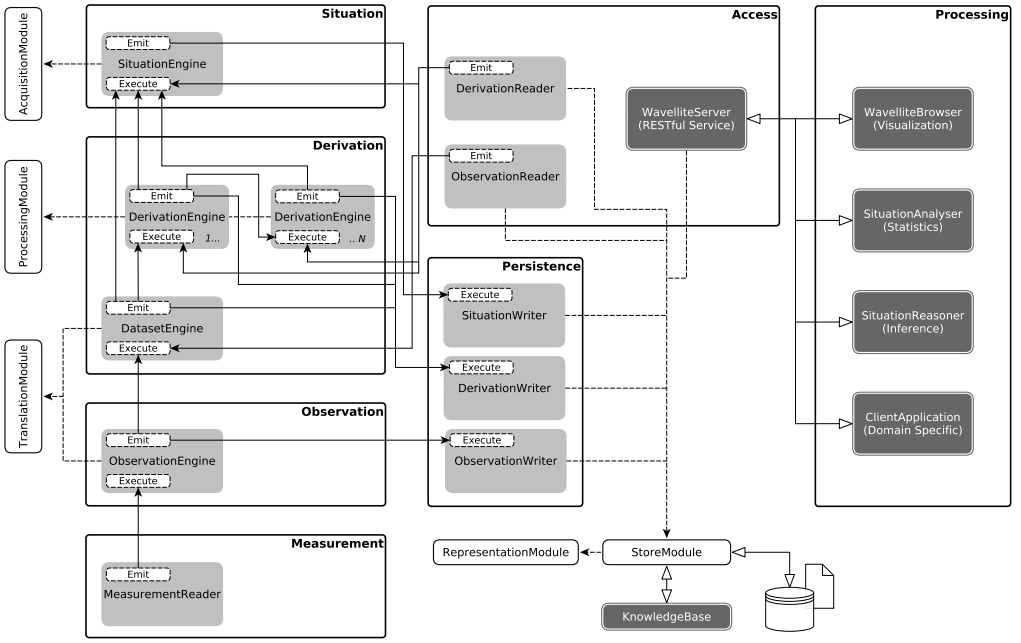


Figure 3.8: The Wavellite computational objects partitioned into its layers.

At the measurement layer, the relevant information object is the measurement result. The relevant information action type is to perform measurement. In Wavellite, measurement results are not stored. Instead, they are necessarily forwarded to the observation layer of the Wavellite application.

At the observation layer, the relevant information objects are the measurement result and the sensor observation. The relevant information action type is the translation of measurement results into sensor observations (Function 3.4). The action type is performed on obtained measurement results. The result of the action are sensor observations.

At the derivation layer, the relevant information objects are the sensor observation and the dataset observation. There are two relevant information action types. First, the translation of sensor observations into dataset observations (Function 3.5). This action type is performed on obtained sensor observations. The result are dataset observations. The second information action type is the processing of dataset observations (Function 3.6). This action type is performed on dataset observations. The result are dataset observations.

At the situation layer, the relevant information objects are the dataset observation and the situation. The relevant information action type is the acquisition of situations from dataset observations (Function 3.7). The action type is performed on obtained dataset observations. The result are situations.

Computational Viewpoint

The computational viewpoint accounts for the main computational objects in the Wavellite software framework and applications, as well as object interfaces. Computational objects are organized according to Wavellite layers. Figure 3.8 is a schematic overview of the computational objects partitioned into layers, and computational object interactions.

Within the measurement layer, the measurement reader is the component that implements data collection. A measurement reader obtains digital numerical values from one or more sensors. Such data are transmitted from sensors to the measurement reader over a communication channel. Typically it is the measurement reader that collects data from sensors. The measurement reader is thus configured so that it can connect to the sensor and request data. However, sensors may also stream data to the measurement reader. Sensors are generally devices, i.e. instruments. They are sources of digital numerical values resulting from measurement, implemented by devices. The measurement reader associates timestamps and metadata to obtained digital numerical values. Values, timestamps and metadata form measurement results M_r. Measurement results are not persisted; they are forwarded to the observation layer.

Within the observation layer, the observation engine is the component that obtains measurement results and translates measurement results into sensor observations, O_s. The translation of measurement results is implemented by a translation module. The module implements the operator $\mathcal{T}_m$ (Function 3.4). Measurement results are generally streamed to the observation engine by the measurement reader. Sensor observations may be persisted by forwarding them to the persistence layer. Sensor observations are forwarded to the derivation layer.

Within the derivation layer, the dataset engine is the component that obtains sensor observations and translates sensor observations into dataset observations, O_d. The translation of sensor observations is implemented by a translation module. The module implements the operator $\mathcal{T}_o(d, \mathcal{C}_p)$ (Function 3.5). Sensor observations are generally streamed to the dataset engine by the observation engine or by the observation reader of the access layer. Dataset observations may be persisted by forwarding them to the persistence layer. Dataset observations may be processed by (chains of) derivation engines and are (eventually) forwarded to the situation layer. The derivation engine is the component that obtains dataset observations and processes source sets of dataset observations into target sets of dataset observations. The processing of dataset observations is implemented by processing modules. A processing module implements an operator $\mathcal{P}_o(d)$ (Function 3.6) and is typically backed by a computational library, such as Apache Commons Math or JScience.

Within the situation layer, the situation engine is the component that acquires situations from dataset observations. A situation engine orchestrates knowledge acquisition and may associate with one or more acquisition modules. The acquisition of situations is implemented by acquisition modules. An acquisition module implements an operator $\mathcal{A}_s$ (Function 3.7). The acquisition of situations requires the attainment of information objects from dataset observations, the mapping of attained information objects to entities of a conceptual model, and the composition of mapped information objects to situations. An acquisition module is typically backed by a computational model for knowledge acquisition from data. Such models may be data-driven or physically-based and employ third party libraries such as WEKA (Hall et al., 2009) or Esper. Dataset observations are generally streamed to the situation engine by the derivation layer, either by a dataset engine or a derivation engine, or by the derivation reader of the access layer. Situations are persisted by forwarding them to the persistence layer.

Within the persistence layer, the observation writer, derivation writer, and situation writer are the components that obtain sensor observations, dataset observations, and situations, respectively, and associate with a store module to persist the objects. It is possible for writers to associate with multiple store modules. A store module associates with a representation module and a store. The store implements the storage and preservation functionality for data and knowledge. The representation module implements the identification and representation functionality for data and knowledge. In Wavellite applications discussed in Chapter 4 the store is, specifically, a knowledge base or RDF database, and the representation module is for RDF and the ontological framework.

Within the access layer, the observation reader is the component that collects sensor observations from the store or external sources. The derivation reader is the component that collects dataset observations from the store or external sources. The observation and derivation readers import data into a Wavellite application. The Wavellite server implements a RESTful web service API (Fielding, 2000) and is the component primarily intended for the retrieval of situations, subsequently processed, e.g. for visualization. However, the Wavellite server can support the retrieval of sensor observations and dataset observations as well.

Within the process layer, the Wavellite JavaScript browser is a generic client application for the visualization of situational knowledge in time and space. The Wavellite browser retrieves situations via the Wavellite server, and processes situations to extract temporal and spatial locations and visualize situational knowledge in space-time. The process layer also includes domain specific applications for situational knowledge processing developed by the science community of a Wavellite application.

3.4 SUMMARY

We have described the architecture and implementation of the Wavellite software framework for situation awareness in environmental monitoring.

The description builds on the ENVRI-RM+K reference model for knowledge-based environmental research infrastructure. The reference model is arguably a useful foundation for an architectural description of the Wavellite software framework because the model explicitly includes sensors and the science community; is expressive enough to support the modelling of Wavellite data and knowledge life-cycles, data and knowledge information objects and action types, Wavellite layers, components, and modules, as well as agents involved in the specification, development, and execution of Wavellite applications; and enables an alignment of the Wavellite software framework with knowledge-based environmental research infrastructure.

The alignment is interesting because it suggests that it may be possible for exiting environmental research infrastructure to adopt ideas developed in this dissertation in order to evolve from predominantly data-based systems into knowledge-based systems, and possibly situation-aware systems.

The main features of the Wavellite software framework can be summarized as follows:

- It supports the implementation of problems with complex data processing and knowledge acquisition tasks that begin with the digital numbers obtained in environmental monitoring and end with represented knowledge about observed situations.

- It supports the explicit representation, using semantic web technologies, of data, information, and knowledge objects as well as provenance information for these objects, including information about the agents and activities involved in object processing.

- It supports the preservation of data, including original sensor data and datasets generated at each processing step. This feature is critical in environmental research infrastructure because scientists need to retain original sensor data, and generated datasets can be valuable data products.

- It distinguishes data and knowledge. Jajaga et al. (2013) describe a sensor deployment in a river, used to collect values about water quality parameters. Depending on the obtained values, the authors classify *sensors* as clean or polluted. This approach highlights the problem of modelling knowledge about the monitored environment at the sensor layer. It is not the sensor that is clean or polluted but the monitored environment, in this case the river, i.e. a spatial feature and object in situations. Hence the separated modelling of observations and situations in Wavellite.

- It distinguishes sensor data and processed data. Applications often require complex data processing chains on sensor data before data can be utilized in knowledge acquisition. The distinct modelling of sensor observations and dataset observations in Wavellite acknowledges that processed data are different from sensor data. Furthermore, compared to an ontology for sensor data, an ontology for datasets is better suited for the representation of multivariate data. It makes thus arguably sense to adopt different ontologies for the representation of sensor and dataset observations. Hence the distinct modelling of sensor observations and dataset observations in Wavellite.

- It allows for arbitrary data processing and knowledge acquisition. This feature is arguably important in environmental monitoring, and research applications, because of the generally large problem spaces, the numerous available methods in data processing and knowledge acquisition, and the heterogeneity of data and situational knowledge.

- It adopts a hybrid approach to situational knowledge acquisition and processing. According to Ye et al. (2012) "a combination of specification- [deductive] and learning-based [inductive] approaches is required to support successful situation identification in a variety of environments and scenarios." Ye et al. note that "specification-based approaches provide the ability to represent situations and incorporate the rich knowledge and semantics required to reason about them." In contrast, "learning approaches [...] have the ability to analyse raw data, and can thus extract patterns and deal with uncertainty." Janowicz et al. (2015) also underscore "that making sense of data and gaining new insights works best if inductive and deductive techniques go hand-in-hand instead of competing over the prerogative of interpretation."

- It uses the concept of situation as abstraction for knowledge acquired from data. Situations "provide a simple, human understandable representation of sensor data to applications, whilst shielding applications from the complexities of sensor readings, sensor data noise, and inferences activities" (Ye et al., 2012).

- It enables the development of environmental monitoring systems that perform automated and near real-time situational knowledge acquisition and processing.

- It supports the implementation of environmental monitoring systems that perceive, comprehend, and project situations. Such systems are thus situation aware.

Having presented the architecture and implementation of the Wavellite software framework, we now turn to concrete case studies that demonstrate the application of the framework in the development of situation-aware environmental monitoring systems.

4 Applications

We provide an overview of the case studies selected to validate the proposed software process for situational knowledge acquisition from environmental monitoring data, and to demonstrate the generality of the framework. At the core of each case study is the development and evaluation of situation-aware environmental monitoring system applications. Each application builds on the Wavelite software framework. The case studies address problems in three distinct domains, namely intelligent transportation systems in Paper **II**; atmospheric science in Paper **III**; and agricultural science in Paper **IV**. Paper **I** builds the foundations for Paper **II**. Paper **V** describes the ENVRI-RM extension and is thus not an application. The three applications presented in papers **II-IV** were developed using evolving versions of the framework. They are presented here in chronological order. Hence, the application in intelligent transportation systems builds on the oldest version of the framework while the application in agricultural science builds on the most recent. The papers **II-IV** present the applications with the framework version at the time of individual application development whereas this chapter discusses the applications using the framework version presented in Chapter 3. Furthermore, the papers present the case studies in greater detail whereas this chapter only provides an overview, and draws a line through framework and application development over time.

4.1 ROAD TRAFFIC

Road traffic situation modelling is the case study in intelligent transportation systems and is the subject of Paper **II**. The case study builds on earlier related work published in Paper **I** and Stocker et al. (2012b), which we briefly present first.

Paper **I** describes an approach for the detection and classification of vehicles in road-pavement vibration data using supervised machine learning methods. The work was conducted during 2010-2011 as part of a project that aimed at systems for situation awareness in an operational environment monitored using a heterogeneous sensor network including video, acoustic, chemical, and vibration sensors. Paper **I** presents the workflow that was developed to continuously collect road-pavement vibration data from one vibration sensor as well as a stream of video camera images; process the collected data using digital signal processing methods to filter and transform vibration signals in time domain into vibration patterns in frequency domain; build training datasets and study the viability of supervised machine learning methods for the detection and classifi-

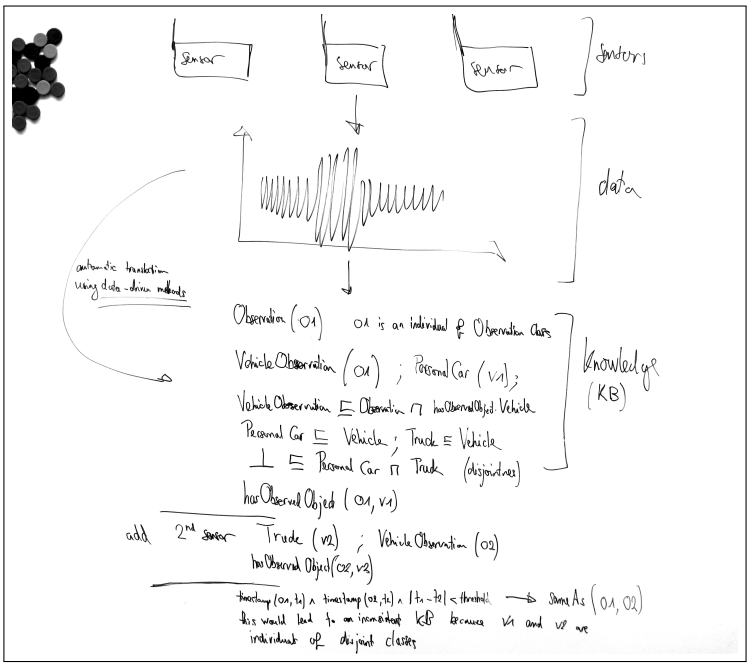


Figure 4.1: Image of the white board showing the sketches for the ideas that formed the foundations of the Wavellite software framework. The approach presented in Paper **I** for vehicle-induced vibration pattern classification served as foundation for the approach to road-traffic situational knowledge acquisition, representation, and processing presented in Paper **II**.

cation of road vehicles by evaluating their performance in classifying vibration patterns. Data was collected for approximately 710 hours, during measurement sessions each lasting roughly a 9-5 workday. The total amount of stored data was approximately 1.7 TB and included 5 billion vibration measurement values. A dataset was created with 1911 labelled vibration patterns, including those for 13 vehicle classes. Vibration pattern classification performance was over 94% for the detection of vehicles, and between 43% and 86% for the classification of vehicles. The emphasis in Paper **I** is on high volume data acquisition and processing, the evaluation of supervised machine learning methods for vehicle detection and classification, and the comparison of our results with those published in related studies. The Wavellite software framework had so far not been conceived.

Stocker et al. (2012b) builds on Paper **I** and utilizes ontology and related technologies to explicitly represent the result of machine learning classification. The approach is interesting in a real-time context for a system that monitors a road section and obtains symbolic knowledge about observed vehicles. The results of road vehicle detection and classification were thus ontology assertions for the class membership of individuals representing vehicles travelling the monitored road section. Stocker et al. (2012b) extended the SSN ontology with domain

knowledge. Observed vehicles were modelled as features, and vehicle properties, such as speed, were modelled as feature properties. The emphasis in Stocker et al. (2012b) is on the explicit representation of knowledge, obtained using data-driven methods from processed sensor data, about physical entities in an environment monitored by a sensor network, and the processing of such knowledge to infer properties of observed entities. We had not utilized the concept of situation. The work so far, however, did pave the road for the development of the Wavellite software framework. As anecdotal side note, Figure 4.1 is a picture dated Saturday, August 27, 2011 of the white board at my office at the University of Eastern Finland, Kuopio campus. The picture shows the sketches drawn on that day for the ideas that formed the foundations of the Wavellite software framework, and the applications discussed next.

In Paper **II**, situations are for observed vehicles travelling a monitored road section. Road-pavement vibration is monitored using three accelerometer sensing devices. Vehicles are detected and classified in road-pavement vibration data using machine learning classification. Specifically, MLP feed-forward artificial neural networks are trained and validated using labelled data. At regular time intervals and for each of the three sensing devices, trained artificial neural networks assess whether a vehicle is observed and whether the observed vehicle is light or heavy. In situations, observed vehicles are said to be *near* a particular sensing device at a certain time. Situations are processed to derive information about vehicle driving side and speed.

The approach proposed in Stocker et al. (2012b), also elaborated for residential building monitoring in Stocker et al. (2012a), has at least two shortcomings. First, in practice sensing devices do not observe vehicles: they observe road-pavement vibration. Second, a system that uses the SSN ontology to represent knowledge about vehicles should not use the same ontology to represent sensor observations for road-pavement vibration. Doing so is hardly elegant as it mixes different levels of abstraction. It soon became thus clear that knowledge about the monitored environment ought to be represented using a different ontology.

Paper **II** addresses these shortcomings. It utilizes the concept of situation, as developed in situation theory, and the STO to represent situational knowledge about the monitored road section. By adopting the STO for the representation of knowledge about structured parts of monitored reality, the approach pursued in Paper **II** permits using the SSN ontology to represent sensor observations. As a consequence, in Paper **II** the feature of interest is the road pavement, and no longer the vehicle, and vibration is the observed property. Sensors thus observe pavement vibration, which reflects the reality on the ground more accurately. As the main contribution, Paper **II** introduces the situation layer, and proposes a first architecture of the Wavellite software framework.

In order to implement the application, the framework is extended with domain knowledge and program logic. Domain knowledge includes the type of

sensing device, i.e. accelerometer; the vehicle types, i.e. light and heavy; the disjointness of light and heavy vehicles; and assertions for infon relations, measured feature and property, and individuals for the three accelerometer sensing devices.

At the measurement layer, three measurement readers (one for each of the three sensing devices) retrieve road-pavement vibration data via HTTP from associated sensing devices. Collected data are binary encoded in Waveform Audio File Format. Binary encoded data are processed to measurement values v_m, timestamps and metadata are composed to contexts $c(v_m)$ of measurement values, and the resulting measurement results M_r are forwarded to the observation layer. At the observation layer, measurement results are translated into sensor observations O_s by an observation engine. Sensor observations are not persisted; they are forwarded to the derivation layer.

At the derivation layer, a dataset engine translates sensor observations into dataset observations O_d of three distinct datasets, d_i, one for each sensing device, $i = 1, 2, 3$. Dataset observations are ordered in time and can be plotted as time series. Datasets are processed using filtering and Fourier transform to enhance the vibration signal induced by vehicles, and to transform signals in time domain into patterns in frequency domain. This transformation is performed every second for a window of length 16 384 of the most recent dataset observations (roughly the past 8 s) by a chain of three derivation engines. The first derivation engine splits datasets d_i into datasets d_i^j of size 16 384, $j = 1, \ldots, k$, where k is the runtime of the experiment in seconds. For datasets d_i^j, the second derivation engine performs a bandpass filter between 100 Hz and 160 Hz to enhance the vibration signal induced by vehicles. The third derivation engine transforms filtered datasets d_i^j from time domain into frequency domain using Fourier transform. The resulting dataset observations are finally forwarded to the situation layer. Dataset observations are not persisted.

At the situation layer, a situation engine obtains Fourier transformed dataset observations, and associates with an acquisition module that classifies dataset observations to assess whether a vehicle is observed by sensing device i within the 8 s interval. Vehicle detection returns class labels, either 'vehicle' or 'no-vehicle'. Such labels are attained information objects. Attained information objects for 'vehicle' are mapped to an individual, ψ, instance of the ontology class `Vehicle`. The result are mapped information objects `Vehicle(`ψ`)`. For detected vehicles `Vehicle(`ψ`)`, vehicle classification returns additional class labels, either 'light vehicle' or 'heavy vehicle'. Such labels are attained information objects. They are mapped to ontology classes, either the ontology class for light vehicle or the one for heavy vehicle. The result are mapped information objects `LightVehicle(`ψ`)` or `HeavyVehicle(`ψ`)`. Situations result from composing mapped information, in particular composing individuals ψ, the `near`-relation,

and individuals for sensing devices, temporal locations and spatial locations to infons of situations. Situations are forwarded to the persistence layer.

At the persistence layer, a situation writer obtains situations and instructs the associated store module to persist situations. The store module uses an RDF representation module to represent situations as sets of RDF statements, and implements persistence using the Stardog RDF database (knowledge base).

Given situations for vehicles being near sensors at certain time points, the application infers infons for the driving side and driving speed of vehicles. This is performed by domain program logic implemented as software agent of the processing layer. The software agent periodically retrieves (recent) situations and performs reasoning. First, it infers the equivalence of vehicles in different situations. Vehicles are inferred to be the same physical object if they were observed by multiple sensors within a short time interval. For such vehicles, the software then checks the consistency of the vehicle type, independently assessed by the three classification processes associated with the three sensing devices. An inconsistency occurs for vehicles inferred to be same but assessed to be of different type. Inconsistencies can be manually reviewed. Finally, for vehicles with consistent vehicle type the software infers infons for vehicle driving speed and driving side.

This first application demonstrates the basic principles for how an environmental monitoring system—consisting of a monitored physical environment, hardware, software, and human agents—can utilize situation theory to model observed situations, and utilize ontology and related technologies to represent situational knowledge obtained from data processed by means of computational models, specifically data-driven models and data acquired from an environmental sensor network.

Though the application relies on several data processing steps, the system architecture presented in Paper **II** does not include the derivation layer. As a result, the system is unable to represent (and curate) intermediate results in data processing, such as filtered and Fourier transformed dataset observations.

The curation of sensor observations and intermediate results in data processing, as dataset observations, is not of primary interest in this case study, for two reasons. First, as the three accelerometer sensing devices each operate at 2000 Hz sampling frequency, the volumes of data are relatively large. Second, sensor observations and dataset observations are arguably of little interest to an application that aims at providing a near real-time picture of situations occurring at a monitored road section. In this case it is situational knowledge that is of greatest interest and worthwhile curating. Data collected in real-time is directly processed and discarded.

In addition to introducing the situation layer, and present its role with an application, a key aspect of Paper **II** is to demonstrate reasoning functionality on situational knowledge curated by the system. Compared to the original data

from which it is acquired, situational knowledge in symbolic form is relatively straightforward to manipulate at the processing layer.

The road traffic case study falls into civil engineering more than environmental science. The two case studies described next are closer to environmental science, and research infrastructure. However, the road traffic case study underscores that the focus of this dissertation is not to develop environmental research infrastructure but to develop a method for situational knowledge acquisition from processed data using computational models, the representation and processing of situational knowledge, and to validate the method for heterogeneous applications. Approaches for situational knowledge acquisition, and its representation, are arguably of interest also to domains other than environmental science but similarly utilize sensor networks to monitor structured parts of reality.

4.2 PARTICLE FORMATION

Atmospheric new particle formation is the case study in aerosol science and is the subject of Paper **III**. The case study builds on earlier related work published in Stocker et al. (2013).

Situations are for events of new particle formation and for cloud events occurring at the Puijo hill in Kuopio, Finland. The application introduces the derivation layer, and thus the support for the representation of intermediate results in data processing. It also demonstrates the application of the software process for the representation of situational knowledge acquired from data to a field of research in science. In addition to software engineers, the application includes a science community consisting of researchers who study atmospheric events such as new particle formation to advance our understanding of the physics underlying the events, and their impact on the climate and human health.

The data are by two sensing devices, namely a Differential Mobility Particle Sizer (DMPS) and a Present Weather Sensor (PWS), and for three properties. The DMPS observes particle size distribution of poly-disperse aerosols and the PWS observes visibility and precipitation, for a volume of ambient air and over time. Situational knowledge for new particle formation events is acquired from DMPS data using machine learning classification. For each day, DMPS data are processed and classified, using trained artificial neural networks, to determine the presence and strength of new particle formation during the day. Situations for new particle formation (`npf`) events support a `npf`-relation infon with objects for event strength and temporal location. Situational knowledge for (rainy) cloud events is acquired from PWS data. A situation for cloud event occurs when average hourly visibility drops below 200 m, and a cloud event is rainy if average hourly precipitation exceeds $0.2\,\mathrm{mm\,h^{-1}}$. Situational knowledge for (rainy) cloud events is acquired from data using complex event processing. Situations for

cloud events support a `cloud-event`-relation infon with objects for the start and end times of the event, and mean visibility. Situations for rainy cloud events support a `rainy-cloud-event`-relation infon with an additional object for mean precipitation.

The application processes historical data, available in text files, between May 2007 and December 2011, of which 2007-2010 is used to train data-driven models for classification. PWS data are available as generated by the sensing device. Thus, the application processes PWS data at the measurement layer, translates measurement results into sensor observations at the observation layer, and persists sensor observations. In contrast, DMPS data are available preprocessed and are thus processed directly at the derivation layer.

At the derivation layer, sensor observations for visibility and sensor observations for precipitation are translated into dataset observations of corresponding datasets. A derivation engine merges the two datasets by processing dataset observations for visibility and dataset observations for precipitation with matching temporal location to dataset observations with three component properties for temporal location, visibility, and precipitation. Furthermore, a derivation reader imports daily DMPS data from text files to daily datasets $m \times n$, whereby m is the number of daily samples in time and $n = 41$ is the number of dimensions, including temporal location and 40 discrete particle diameter sizes. A derivation engine processes the observations of daily datasets between 6 AM and 6 PM to a (single) dataset observation using Singular Value Decomposition (SVD). Dataset observations generated at the derivation layer are persisted.

At the situation layer, a situation engine obtains dataset observations for visibility and precipitation, and associates with an acquisition module for complex event processing to identify time intervals $[t_1, t_2]$ during which visibility is continuously below 200 m. Mean visibility, v, and mean precipitation, p, during $[t_1, t_2]$ are computed. The values t_1, t_2, v, and p are attained informations objects. The objects t_1 and t_2 are mapped to individuals (OWL-Time) `Instant(`t_1`)` and `Instant(`t_2`)`, and the objects v and p are mapped to individuals (STO) `Value(`v`)` and `Value(`p`)`. Mapped informations objects are finally composed to situations. If $p \leq 0.2\,\mathrm{mm\,h^{-1}}$, situations support a `cloud-event`-relation infon with objects `Instant(`t_1`)`, `Instant(`t_2`)`, and `Value(`v`)`. If $p > 0.2\,\mathrm{mm\,h^{-1}}$, situations support a `rainy-cloud-event`-relation infon with the additional object `Value(`p`)`.

A second situation engine obtains dataset observations with SVD-processed daily particle size distribution data, and associates with an acquisition module that classifies dataset observations using MLP artificial neural networks to assess the presence of a new particle formation event during the day and, if present, the strength of the event. The class label for event strength, c, is an attained information object. It is mapped to an individual `Value(`c`)`. Mapped information is composed to situations that support a `npf`-relation infon with objects `Value(`c`)` and `Interval(`t`)` for the day t at which new particle formation is identified.

Situations are forwarded to the persistence layer where a situation writer requests the associated store module to represent situations in RDF and persist RDF statements to the knowledge base. For the year 2011, the application acquired 45 situations for new particle formation events and 126 situations for cloud events, of which 52 were rainy.

Paper **III** introduces the derivation layer and the use of the QB vocabulary to differentiate sensor observations from dataset observations, i.e. raw sensor data from processed (sensor) data. It is therefore possible to model data derivation and to represent and curate derived data products. The architecture presented in Paper **III** has been stable over time and other applications. In fact, it is reflected in Figure 3.8 which, compared to the architecture in Paper **III**, encapsulates certain components into the access and persistence layers but leaves the core of the architecture unchanged.

The case study discusses several products obtained in situational knowledge processing. Situations of new particle formation for 2011 are plotted to show when the events occur during the year and how strong they are. The total number of events per event strength class is calculated. Situations of (rainy) cloud events for December 2011 are plotted to show when the events occur during the month, how long they last, and indicate mean visibility and precipitation (if applicable). The longest cloud event, the cloud event with lowest mean visibility, the rainy cloud event with maximum mean precipitation during 2011 and December 2011, in particular, are calculated. The analysis of situations on curated situational knowledge is often trivially achieved with little more than a SPARQL query, and generally more straightforward than on data.

The products resulting from situation analysis, such as plots, are intended for human consumption. Plotting is indeed widely used to present data and information to humans, and researchers in particular. Presented with a figure during a talk, a group of experts can with little effort obtain information which the speaker intends to convey to the audience. For instance, experts can easily tell that in 2011 most new particle formation events were weak. This is possible for experts because they combine relevant contextual information with visualized data to obtain implicit knowledge. Relevant contextual information in this case includes that the speaker is presenting data for the strength of new particle formation events during 2011, and that the strength of new particle formation events is classified into strong, intermediate, and weak by the numbers 1, 2, 3, respectively. However, implicit information in figures is hardly accessible to machines. With Wavellite, situational knowledge is explicit, represented using machine readable and interpretable statements, and thus accessible to computer systems.

Situational knowledge for new particle formation automatically assessed by a computer system is useful to aerosol scientists of the science community. Aerosol scientists, who collaborated with software engineers in this case study, indi-

viduate and assess the strength of new particle formation in daily DMPS data manually. Matlab scripts are used to plot the data, and new particle formation is analysed visually. The results of such analysis, including the strength of new particle formation, is recorded in Excel. Clearly, the automated assessment by a computer system can support the science community and streamline the workflow, which is simplified to a mere review of the automated assessment. Furthermore, knowledge about situations of new particle formation is recorded in a knowledge base that supports rich semantic descriptions, and the retrieval and discovery of knowledge.

4.3 PLANT DISEASE PRESSURE

Plant disease pressure situation modelling in agriculture is the case study in agricultural science and is the subject of Paper **IV**. During growing seasons, agricultural advisers to farmers assess and monitor crop disease pressure in agricultural parcels. Disease is caused by pathogens. Agricultural parcels are land areas on which farmers grow crops.

Situations of primary interest are forecast outbreaks or acute outbreaks of pathogens in crops. Hence, situations support an `outbreak` or `acute-outbreak`-relation infon with objects for the pathogen, temporal location, and spatial location. The time intervals during which situations occur are temporal locations. The agricultural parcels are spatial locations. The case study considers 3 fungal pathogens, 2 cereal crops, and 17 agricultural parcels. The agricultural parcels are located in Finland.

The application uses a disease pressure model to assess (acute) outbreak situations. The model computes the (daily) accumulated risk of disease in the crop of an agricultural parcel using weather forecast data for temperature, relative humidity, wind speed, and precipitation amount. The 2-day forecast weather data are obtained from the Finnish Meteorological Institute (FMI) for a region that spatially overlaps the agricultural parcels. The weather data are retrieved using the FMI Open Data web service. FMI weather forecast data are modelled data resulting from a computational model, not observational data resulting from sensor measurement. Thus, in this application weather data for temperature, relative humidity, wind speed, and precipitation amount are dataset observations.

The science community in this case study consists of agricultural advisers and software engineers. Agricultural advisers are domain experts and provide contextual information about the fungal pathogens, cereal crops, and agricultural parcels. They also developed the disease pressure model. The model is implemented by software engineers, who also extend the Wavellite framework with domain knowledge and program logic in order to implement the application. The application is executed for the 2014 growing season.

At the derivation layer, each day a derivation reader retrieves forecast data from FMI and processes the NetCDF (Rew and Davis, 1990) encoded data returned by FMI to dataset observations. Dataset observations for weather forecast data are forwarded to a derivation engine, which uses the disease pressure model implementation to compute the accumulated risk for each agricultural parcel and (applicable) pathogen. The result of the computation by the derivation engine are dataset observations with information for time, agricultural parcel, pathogen, and accumulated risk. Such dataset observations are elements of the disease pressure dataset. Disease pressure dataset observations are forwarded to the situation layer. Dataset observations are also forwarded to a derivation writer of the persistence layer, which requests a store module to represent dataset observations in RDF and persist the RDF statements to the knowledge base.

At the situation layer, a situation engine acquires situational knowledge for (acute) outbreak situations from disease pressure dataset observations. A situation of outbreak for a pathogen in an agricultural parcel occurs (or persists) when the accumulated risk is within the interval $]50, 75]$. A situation of acute outbreak for a pathogen in an agricultural parcel occurs (or persists) when the accumulated risk is > 75. Thus, situational knowledge acquisition from dataset observations is relatively trivial in this application as it merely amounts to testing the computed accumulated risk for two thresholds. Which threshold the accumulated risk exceeds specifies the infon relation used in situations. Composed are also the pathogen, temporal location, and spatial location; they are mapped information objects related to disease pressure dataset observations. Situations are forwarded to a situation writer of the persistence layer, which requests a store module to represent situations in RDF and persist the RDF statements to the knowledge base.

The Wavellite server and the Wavellite browser are introduced in this case study to support agricultural advisers at the Natural Resources Institute Finland in the monitoring of situations of (acute) outbreaks as projected by the system during the growing season. The Wavellite server is an Apache Tomcat web application and implements a RESTful API. It is a component of the access layer. The version developed for this application supports the retrieval of situational knowledge encoded in plain text, RDF, or JSON. The Wavellite browser is a generic JavaScript client application for situational knowledge visualization. Given that situational knowledge is generally located in space-time, the Wavellite browser uses time line and map visualization for situational knowledge. In other words, situational knowledge is visualized in time using a time line and in space using maps. The Wavellite browser interacts with the Wavellite server and is a component of the processing layer.

In this application, situations for (acute) outbreak are visualized for their duration on the time line, using different colours for outbreak (orange) and for

acute outbreak (red). Such colour customization can be configured by mapping infon relations to colours. The spatial locations at which situations occur (at any time) are highlighted on the map. Elements on time line and map can be selected to obtain more information about situations. Additional information includes the pathogen name, time period, agricultural parcel name, and whether it is a situation of outbreak or acute outbreak.

The Wavellite browser is designed to be generic. Given a set of situations, the browser visualizes situational knowledge along temporal and spatial dimensions. This is arguably straightforward situational knowledge processing. However, the ease is partially due to situations sharing a common vocabulary, i.e. the one specified by the ontological framework centred around the STO. The browser is implemented against this ontological framework, which specifies the syntax and semantics of situational knowledge. Given a set of situations, the browser can thus expand on situations, infons, infon relation and objects to obtain temporal and spatial locations of situations, used to visualize situations on the time line and map of the Wavellite browser. The ontological framework thus serves as a specification for the information objects shared between the Wavellite server and the Wavellite browser.

The data and information used in this application are sourced from heterogeneous digital documents, including Excel and ArcGIS files as well as web services. Excel files describe the characteristics of agricultural parcels, e.g. information about crops. ArcGIS files contain the polygon data of agricultural parcels. Web services provide weather forecast data. The application integrates the heterogeneous data so that it conforms to the syntax and semantics of the ontological framework. For instance, the polygon data of agricultural parcels provided by ArcGIS files are translated into Well-Known Text for the textual representation of GeoSPARQL geometries, associated with GeoSPARQL features (i.e. the agricultural parcels). These features are used as values of dataset observation component properties and objects of infons in situations. Similarly, forecast weather data obtained via the FMI Open Data web service are translated into values of dataset observation component properties. As a result of the integration, data, information, and knowledge is encoded in RDF, and SPARQL can be used to, for instance, query for the pathogens involved in outbreak situations lasting at least three weeks at agricultural parcels of a given area where farmers grow wheat.

The application employs a disease pressure model to compute the accumulated risk of disease from forecast weather data. It is an environmental (ecological) physically-based model and it stands in contrast with the computational models used in the other applications. First, the road traffic and particle formation applications rely on data-driven models, predominantly artificial neural networks. Second, the disease pressure model is used by a processing module at the derivation layer whereas the artificial neural networks are used by an acqui-

sition module at the situation layer. Heterogeneous computational models can thus be utilized for various purposes and can play a role at different layers of the architecture. Furthermore, computational models utilized in applications can be of various type, in particular data-driven and physically-based.

Finally, Paper **IV** grounds the Wavellite software framework in situation awareness. The acquisition—from processed environmental monitoring data—and the representation and projection of situational knowledge for (acute) outbreaks of pathogens in agricultural crop is argued to be a process of situation assessment performed in order to obtain and maintain situation awareness. Situation awareness is distributed among the technical components of the situation-aware environmental monitoring system and the social components of the system, specifically agricultural advisers and farmers.

4.4 REMARKS

We have presented the three main applications developed as part of the dissertation to validate the proposed software process for situational knowledge acquisition from environmental monitoring data, and the architecture and implementation of the Wavellite software framework, for different case studies. The chapter has briefly summarized the three situation-aware environmental monitoring system applications in case studies for road traffic, particle formation and cloud events, and plant disease pressure situation awareness.

The three case studies are distinct in the environmental phenomena, objects in situations; in the utilized data, including observation data and model data; in the methods used for data processing and knowledge acquisition, encompassing digital signal processing, complex event processing, machine learning and physically-based modelling; and in the role layers, components, and modules of the architecture play in different applications, as they only utilize required functionality. However, the three applications share the problem of situational knowledge acquisition from data and the representation of situational knowledge. Data results in environmental monitoring and knowledge is for situations with environmental phenomena as their objects.

Situation-aware environmental monitoring systems consist, in general, of various agents: hardware, software, and people. The systems presented here utilize situation theory to model observed situations, and utilize ontology and related technologies to represent situational knowledge obtained from data processed by means of computational models. The three applications thus validate the claims **C1** and **C2**, and thus provide positive evidence for the research question.

In order to briefly discuss forms of reasoning other than classical ontology reasoning discussed in Paper **II**, we highlight a fourth application for its use of spatial reasoning in situational knowledge processing. Discussed in detail

in Stocker et al. (2015a), the application demonstrates how situational knowledge for the location of storms and the location of drivers in space-time can be utilized to discover situations in which drivers may be at higher risk because they are (forecast to be) located within storms. Situational knowledge for storms and their spatial extent is obtained from radar data for the reflectivity of rainfall intensity. Storms are objects in situations. Similarly, drivers and their spatial location in time along routes are objects in situations. Driver locations are computed from data for the origin, destination, and departure time provided by drivers. The system uses the Google Directions API to obtain a route and the expected arrival time. The application uses the Profium Sense RDF database which, in contrast to the Stardog RDF database, at the time of writing supports quantitative and qualitative spatial reasoning. This feature is utilized in SPARQL queries to discover situations in which drivers may be at higher risk as they are located within storms. Discovered situations can be represented explicitly. Other than for its use of spatial reasoning in situational knowledge processing, this application is also different from the other applications because it employs two distinct systems that independently acquire situational knowledge: one based on radar data provided by FMI and the other based on user input and the Google Directions API. Due to their commitment to represent situational knowledge conforming to the ontological framework centred around the STO, it is straightforward for a system to integrate situational knowledge for storms and situational knowledge for drivers, and formulate the discovery of new situations as a SPARQL query.

The proposed approach to situation-aware environmental monitoring systems is arguably interesting also to domains and problems other than those presented. For instance, intelligent systems in smart homes and smart cities can communicate knowledge about current and projected situations rather than presenting users with measurement values of one or more sensors. Stocker et al. (2012a) discussed the problem for a smart home and situations of unhealthy exposure to carbon monoxide in indoor air, a composite concept for situations in which inhabitants of a residential home are exposed to a dangerous gas for durations and concentrations exceeding defined thresholds.

The discussed applications have in common that knowledge acquired from data is mapped to knowledge base *assertions*. This is because situations involve individual environmental phenomena and are generally located in space-time. Situational knowledge is therefore assertional knowledge, as it states the concept and role assertions of individuals. In other words, the described situation-aware environmental monitoring systems populate the ABox (assertional box) of knowledge bases.

An environmental monitoring system can also obtain terminological knowledge from data. This is exemplified in Stocker et al. (2011) where the authors learn from data the threshold value t of a rule atom a that requires the variable v

to exceed the threshold, i.e. $v > t$, whereby the atom a is part of the antecedent $a \wedge b$ of a rule $a \wedge b \rightarrow c$. Rules are terminological knowledge and part of the TBox (terminological box) of a knowledge base. In Stocker et al. (2011) the data are for the measured nutrient concentration in lakes. Clustering is used to separate lakes rich in nutrients from those poor in nutrients. The two centroids resulting from clustering represent a central tendency for the concentration of nutrient rich lakes and the concentration of nutrient poor lakes. The method thus computes the mean of the centroids. The mean is the threshold value t. If the threshold is exceeded then a lake is nutrient rich; otherwise the lake is nutrient poor. Given an individual lake with measured nutrient concentration, the rule determines whether the individual lake is nutrient rich or poor.

Together with related work, the presented applications draw a line through the evolution of the Wavellite architecture, from early applications in which knowledge about observed environmental phenomena is mapped to entities of the SSN ontology, to first applications that model observed environmental phenomena as objects in situations and thus adopt the STO, and following applications that include the explicit representation of datasets using the QB vocabulary. The Wavellite architecture was thus gradually refined and improved with new features as the development of applications brought to light new requirements. The evolution underscores a departure from building Wavellite applications on top of the Apache Storm real-time computation system, an approach that was pursued in early applications. The development of applications has relied on interactive workflow, as data are retrieved from the store in order to develop data processing and knowledge acquisition modules. For real-time applications, Apache Storm or similar computation systems are arguably interesting. However, not all applications execute exclusively in real-time. In particular, environmental research infrastructure also relies on interactive workflow. Thus, systems may want to support both the interactive and the streamed operation modes.

5 Discussion

Having presented the Wavellite software framework implementation and the applications we developed using the framework, this chapter discusses the work. Section 5.1 quotes various authors who have highlighted the 'semantic gap' between low-level sensor data and high-level knowledge, and the difficulty of closing this gap. The section also presents various existing software architectures that aim at addressing this issue. We highlight the main distinguishing features between Wavellite and the presented related software architectures. Section 5.2 discusses alternative theories for the concept of situation and systems that utilize situation as abstraction for high-level knowledge. Section 5.3 discusses theories for the concept of event and systems that utilize event as abstraction for high-level knowledge. We note that, never mind the chosen abstraction, the surveyed systems share the goal of 'closing the semantic gap' with Wavellite. Section 5.4 presents systems for situation awareness. We underscore the applicability of situation awareness to environmental monitoring and the key differences between situation-aware environmental monitoring systems developed with Wavellite and situation awareness systems in more traditional domains, such as aviation. Section 5.5 discusses sensor data management, related issues and systems. We highlight that situational knowledge acquisition, curation, and processing are characteristics that distinguish Wavellite from the reviewed informatics platforms for sensor data management. Section 5.6 provides an overview of various domains, such as autonomous robotic systems and ambient intelligence, in which obtaining symbolic descriptions about an environment perceived using sensors is a challenge, too. The purpose of this section is to highlight that the fundamental problem addressed by this dissertation is shared with other domains in which systems employ sensing devices to perceive an environment. Sections 5.7 and 5.8 discuss the main strengths and limitations of our work, respectively. We conclude with remarks on future work in Section 5.9.

5.1 SENSE MAKING

The problem of 'making sense' of data acquired from sensor networks—and environmental sensor networks used in environmental monitoring for scientific applications, in particular—is a widely recognized research problem, and motivated this dissertation.

Underscoring the growing need to analyse sensor data, Cook (2007) highlights that to make "sense of sensor data is a complex task" and notes that the large volumes of multidimensional streamed sensor data can hardly be analysed manually. Ganguly et al. (2007) note that it is critical for scientific applications

"to generate insights or new knowledge from sensor data." Balazinska et al. (2007) argue that the 'worldwide sensor web' "must incorporate logical data abstractions and visualizations that can shield users from the complexities of the underlying sensing infrastructure." Looking "at the flood of collected and integrated real-time sensor data," it is clear to Nittel et al. (2008) "that the cognitive aspects of users must be addressed and that higher-level, semantically rich data representation models and query languages are necessary." Discussing mobile and pervasive computing scenarios, Castelli et al. (2009) highlight that "there is a huge gap between low-level sensor readings and high-level situation awareness." Writing about NEON, Tollefson (2011) quotes Sandy Andelman, an ecologist with Conservation International, who predicts that "to manage and process and make sense of [NEON] data is going to be a huge challenge." Conroy et al. (2011b) highlight the 'semantic gap' between sensor data and expert information needs in the context of athletes monitoring during training activities. Heintz et al. (2010) state that the "gap between sensing and reasoning is quite wide, and cannot in general be bridged in a single step." Barnaghi et al. (2012) concur by stating that the "[e]fficient use of [...] sensor data involves making sense of massive amounts of data in order to convert it into information and knowledge from which humans can gain insights and base decision." Fiorini et al. (2013) comment that the "huge amount of acquired [sensor] data [...] requires more robust approaches to filter and organize the information in order to support decision making," and note that "it is difficult to conciliate the low abstraction level of the raw data with the information structure needed by intelligent agents in order to make decisions and interpretations." Referring to the Internet of Things, Alirezaie and Loutfi (2014) note that "[a]s more sensors of varying modality become connected, it will be of importance to provide automated interpretation of the sensor data." Discussing the problem for mobile users and devices, Sarma et al. (2014) state that the "[e]ffective use of the sensed data relies on effective 'sensemaking' that transforms the gathered data to meaningful information for improved situational awareness, decision making and control." Sarma et al. underscore that making sense of sensor data is a challenging task and new architectures that address the issue are needed. Referring to large volumes of data from wireless sensor networks, Roda and Musulin (2014) state that "the need for computer-aided systems that extract useful knowledge [from such data] becomes evident." Reviewing situation identification techniques in pervasive computing, Ye et al. (2012) conclude that a pervasive computing system should interpret sensor data into domain-relevant concepts. In their example, data for heart rate or blood pressure should be interpreted so that the pervasive computing system 'knows' "whether the user is suffering a heart attack or exercising." Despite rapid increase of (streamed) data available on the web, such as for weather forecasts or traffic directions, Pongpaichet et al. (2013) note that "comprehensive development [of] tools and computational frameworks for ef-

fectively combining and processing these available heterogeneous streams are lacking." Reviewing the state of the art of automated scene interpretation from aerospace sensors, Herbin et al. (2012) state that "[t]he role of scene understanding [...] is to generate a formal description that can be communicated, stored or enhanced by various agents, either artificial or human." Herbin et al. continue noting that "[h]umans have no difficulty in describing what they see in an image [...] and in reasoning about the cause," a capability not found in computers. The authors refer to the expression 'semantic gap' which "expresses the fact that the information encoded in computers does not spontaneously match the inner structure of sense-data." Wetz et al. (2014) state that the "[a]vailability of raw [sensor] data can only be a first step, which has to be followed by enrichment with contextual information and careful processing to extract relevant insights." The authors also note that "[m]eans to exploit the continuously generated data, however, are still scarce."

Accordingly, the problem has been approached by various authors. Liu and Zhao (2005) and Whitehouse et al. (2006) present Semantic Streams, a framework that allows "users to pose queries over semantic interpretations of sensor data, such as 'I want the ratio of cars to trucks in the parking garage', without actually writing code to infer the existence of cars or trucks from the sensor data." In other words, "instead of querying raw magnetometer data, the user queries whether vehicles are cars or trucks." The key to the framework are inference units, i.e. processes (applications) that operate on event streams. Event streams flow through combinations of inference units. Events represent observed objects and their properties, such as a detected vehicle and properties for time, location, speed. Inference units can obtain semantic information about an observed environment from event streams and generate new event streams or enrich events with new properties. Semantic Streams uses a logic-based markup language to support the description of inference units, i.e. their input and output streams and relationships between them. For example, "a vehicle detector unit could be described as an inference unit that uses a magnetometer sensor to detect vehicles, and creates an event stream with the time and location in which the vehicles are detected." Queries are first-order logic descriptions of event streams and properties. For instance, a user query may ask for events that are cars in a certain region. Given such a query, Semantic Streams attempts to compose inference units to answer the query. It is possible that a query cannot be answered, in which case the system would have to be extended with new sensors or new inference units. Query evaluation builds on the standard backward chaining algorithm, whereby query predicates are matched with the consequent of a rule, thereby triggering the matching of rule antecedents, or with a fact in the knowledge base.

Gaglio et al. (2007) present a 'cognitive architecture' designed to extract information about the environment from raw data collected by a wireless sensor

network. The architecture was developed for artificial vision (Chella et al., 1997), adapted to wireless sensor networks, and extended (De Paola et al., 2009) in the context of Ambient Intelligence (Remagnino and Foresti, 2005, AmI). The architecture proposed by Gaglio et al. consists of three layers: subsymbolic, conceptual, and symbolic. The subsymbolic layer involves sensors that collect raw data for monitored properties of an observed environment and performs preliminary data processing, which to some extent may be performed by the sensor network. At the conceptual layer, processed data are described as vectors in a conceptual space (Gärdenfors, 2004). A vector is a point and is called 'knoxel'. A conceptual space is a set of quality dimensions, e.g. temperature and mass, or a set of spatial dimensions. It is a metric space in which the similarity of knoxels can be defined. At the symbolic layer, the architecture produces descriptions about the monitored environment in terms of a logical language. This is achieved by mapping knoxel sequences onto assertions, i.e. by mapping structures of conceptual spaces onto symbolic constructs. Symbols at the symbolic layer are grounded in the conceptual layer.

Ganguly et al. (2007) propose a framework for knowledge discovery on environmental data in scientific applications, whereby the data are sourced from sensors or repositories and are primarily intended for natural disaster early warning systems. The framework consists of offline and online parts. The offline subframework for predictive analysis integrates data from heterogeneous sources, including remote sensors and *in situ* sensor networks, model output, and domain knowledge. Integrated data serves in pattern and process detection, performed offline using techniques in signal processing, data mining, statistics. The results build a body of actionable knowledge. The online sub-framework for decision making performs (near) real-time analysis of model and observation data and utilizes the knowledge obtained from offline analysis to "facilitate short-term decisions and longer-term policies."

Heintz et al. (2010) propose a conceptual software framework for modelling knowledge processing applications, called knowledge processing middleware. The software framework is designed to bridge the gap between sensing and reasoning in a physical agent, and is discussed for traffic monitoring by autonomous unmanned aerial vehicles. The authors underscore that "[b]ridging this gap is a challenging problem" and doing so "in a single step, using a single technique, is only possible for the simplest of autonomous systems." Heintz et al. propose six design requirements for knowledge processing middleware. First, the middleware should "permit the integration of information from distributed sources, allowing this information to be processed at many different levels of abstraction and finally transformed into a suitable form to be used in reasoning." Wavellite addresses the integration of data and information from distributed sources, in particular sensors and databases, but also files. Integration is achieved by aligning data and information to terms and semantics of an

ontology framework. Wavellite supports processing data at different levels of abstraction, namely at the observation and derivation layers of the architecture. Finally, Wavellite supports the transformation of data into situational knowledge, suitable for reasoning. The second design requirement suggested by Heintz et al. is "to support both quantitative and qualitative processing." Wavellite supports quantitative processing of (sensor) data as well as the representation and processing of qualitative relations among objects in situations. The third design requirement is that "both bottom-up data processing and top-down model-based processing should be supported." As the applications presented in Chapter 4 clearly show, Wavellite is designed to support bottom-up data processing. The applications also demonstrate examples of top-down model-based processing, e.g. in form of situational knowledge processing, such as (rule-based) reasoning. The fourth design requirement is "support for management of uncertainty." This requirement is not addressed in Wavellite. In particular, the framework has so far not investigated the representation of uncertainty at the various layers, and uncertainty propagation between layers. The fifth design requirement is the support for "flexible configuration and reconfiguration of knowledge processing." This requirement is not addressed by Wavellite. Initial Wavellite versions used Apache Storm, and applications were implemented as Storm topologies. Knowledge processing could thus be relatively flexibly configured (and reconfigured) by specifying Storm topologies. However, Storm was later abandoned and currently there is no supported flexible configuration approach. Finally, and also not addressed by Wavellite, the sixth design requirement is "to provide a declarative specification of the information being generated and the information processing functionalities that are available."

Moodley and Tapamo (2011) argue that sensor systems should "support knowledge capture and use" in addition to dealing "with issues around the provision, fusion and analysis of heterogeneous data" and note that "[w]hile it has been acknowledged that abstractions are required to bridge the gap between sensors and applications [...], the most effective mechanism [...] remains an open issue." Moodley and Tapamo present an ontological framework designed to support the representation of theme, space, time, and uncertainty of observations as well as the agents involved in workflow tasks, such as sensors from which data are collected, or algorithms that process data.

Conroy et al. (2011b) discuss the EventSense architecture, "a framework and methodology for automated processing of sensor data so that it can be queried using a standard query language." Conroy et al. develop the system for the sport and health domain where heterogeneous sensor networks are utilized to detect "various biological and physiological properties in athletes during training activities" to identify "key intervals in exercise such as moments of stress or fatigue." The architecture is presented for cycling, and broadly consists of four components: hardware, data management, metadata, and query processing. The data

management component consists of three processors: sensor enablement, contextual enrichment, and integration. Sensor enablement converts heterogeneous sensor data in plain text format to a standard XML format. This conversion process is facilitated by user defined templates. The resulting data in XML can be queried using XML query languages. However, not all user information-needs can be directly translated into XML queries because the data lacks of semantics. The contextual enrichment processor aims at reducing semantic gaps. The processor is provided with event definitions, i.e. user defined descriptions for what action applies under what conditions. Conditions can include complex user defined functions. Actions update data and can enrich data with metadata. For instance, an event definition may state that a terrain is a 'steep climb' section of a race if GPS values match certain conditions. Thus, the contextual enrichment processor annotates data with metadata required in user queries. User information needs may require (enriched) data from multiple sensors. For such queries, "multiple sources of evidence must be integrated." This integration is addressed by the integration processor. The authors have discussed the approach also for knowledge acquisition from sensor data in an equine environment (Conroy et al., 2011a) where horses and jockeys were equipped with sensors in order to identify the most energy demanding events and to classify horse and jockey movement during horse-racing training exercises. Cappellari et al. (2011) restructured the architecture into five modules: hardware, sensor enablement, context enrichment, data transformation, and sensor data storage and query. In contrast to Conroy et al., the Cappellari et al. architecture describes the sensor enablement and contextual enrichment modules in more details and includes data transformation as an additional module for computations, such as rolling average.

Negru (2012) describes SemaKoDe, a system architecture that adopts semantic web technologies "to automatically annotate, reason, classify and operate with sensor data." The architecture consists of five layers: knowledge base, network management, database, discovery, and application. The knowledge base layer manages terminological knowledge. The network management layer is primarily responsible for the collection of data from sensors. The database layer obtains data collected at the network management layer and performs standard knowledge discovery in databases (Fayyad et al., 1996) operations as well as semantic annotation of sensor data, persisted in a triple store. The discovery layer encapsulates data mining and reasoning algorithms. Data mining is performed on processed data while reasoning is performed on terminological knowledge of the knowledge base layer and assertional knowledge of the triple store. Finally, the application layer exposes query endpoints and services. The architecture is briefly discussed for a scenario of fire hazard in a hospital.

Barnaghi et al. (2012) describe a "framework for perception creation from sensor data." The authors stress the importance of machine-interpretable ab-

stractions created from processed data. The framework collects sensor data and transforms sensor data into observations, semantically annotated according to the SSN ontology. This transformation enriches sensor data with qualitative information. For instance, a period of measured temperature greater than 30 °C is transformed into an observation of 'high temperature'. The semantic annotation with qualitative information is achieved by segmenting sensor data into patterns, using the Symbolic Aggregate Approximation algorithm (Lin et al., 2003, SAX). Patterns are then compared to *labelled* patterns using a similarity function in order to derive the annotation for the observation. Given a set of observations, the framework also "determines the best explanation for a set of observations," using domain knowledge and by means of the Parsimonious Covering Theory (Reggia and Peng, 1987, PCT). For instance, given an observation annotated as being 'cold temperature' the framework may determine that the window was left open. Henson et al. (2012) describe how the process of determining the best explanation for observations can be implemented using OWL reasoners, and evaluate the proposed method for a use case in the health care domain. Ganz et al. (2013) improve on various aspects of the framework. First, the SAX algorithm for creating patterns from time series data is optimized for sensor data with variable encoding rate in order to achieve higher compression and better reconstruction. Second, PCT explanations for observations are extended with probabilities. Third, PCT is also extended with a Hidden Markov Model (Rabiner and Juang, 1986, HMM) "in order to infer abstractions based on time-dependent sensor data." Building on this work, Ganz et al. (2014) present a framework that constructs topical ontologies from sensor data and aims at data-driven ontology construction. The framework utilizes clustering to group SAX discretized sensor data, which is achieved by modifying the *k*-means algorithm (MacQueen, 1967) for non-numerical SAX based patterns. Identified clusters represent concepts of the topical ontology, and are unlabelled at this stage. The labelling of clusters is achieved using a rule-based reasoning mechanism. Temporal relations between clusters are obtained using a Markov model.

Highlighting the lack of "a layer specifying the transition from observation data to [ontology] classes and relations" Janowicz (2012) proposes a framework that builds on semantically described sensor observation data, and employs methods in data mining and machine learning to construct ontological primitives, i.e. atomic classes and relations. The mapping between data and primitives is enabled by 'semantic signatures'. Signatures can be mined from data, in which case they represent data patterns over, e.g., temporal and spatial dimensions. Following the example by Janowicz, plotting people density over time and space shows that universities are attended during weekdays and working hours whereas restaurants are visited during weekends and evenings. Thus, the classes for university and restaurant are described by different primitives. Primitives are then integrated following ontology design patterns (Gangemi, 2005).

Myers and Trevathan (2013) present "a framework for integrating remotely sensed data with web-available static data for use in observational hypothesis testing and the analysis phase of research." The framework consists of three main components. First, a wireless sensor network is used to collect data about some properties of a monitored environment. Coupled with this component is an environmental model, which is updated with sensor data and is capable of re-tasking sensors. A second component obtains data from heterogeneous sources, including sensor data, executes a workflow to transform static and dynamic data so that they conform with a hierarchy of ontologies, and persists the results in a knowledge base. This component supports the formulation of hypotheses in form of rules. The third component is a "platform for distributed data sharing and processing that enables researchers, managers and decision-makers to collaborate around the data." This third component is a 'data hub' and addresses the collection, discovery and curation of data. Myers and Trevathan discuss the framework for a uses case in which scientists can investigate the effects of human coastal population density on algal blooms and seagrass.

Gorrepati et al. (2013) present an architecture for a system that utilizes acoustic sensors to monitor bird calls and processes the data in order "to recognize bird calls, identify birds, classify species, and track bird behaviour in a bird's ecological environment." Knowledge, extracted from data, about birds is represented according to an OWL ontology using semantic web technologies. The architecture consists of five layers: physical, event, semantic, awareness, and service. The physical layer is concerned with sensors and signal processing. The event layer processes and classifies data. The semantic layer represents acquired knowledge, such as the bird species and location. The awareness layer enriches knowledge represented at the semantic layer by means of rules to, e.g., infer bird flight direction, interaction, or health. Finally, the service layer consists of service applications, such as for data management.

Alirezaie and Loutfi (2014) present a system that attempts to draw explanations for changes detected in sensor data. The system is presented for a kitchen equipped with a sensor network. A gas sensor monitors ambient air while other sensors monitor motion, luminosity, temperature. Data from the sensors installed in the kitchen are processed to detect changes, which are modelled as events. The system assists the interpretation of events of abnormal odour in air, as detected in gas sensor data, and utilizes ontological knowledge about odours, their causes, and relations to other phenomena. Specifically, the system attempts to infer an explanation for detected abnormal odour, such as cooking or burning. Reasoning is achieved by means of Answer Set Programming (Baral, 2003, ASP), domain knowledge and rules, and contextual information about the state of appliances in the kitchen. Reasoning thus involves a conversion of OWL axioms and assertions into ASP rules. According to the authors, ASP is interesting because it can handle incomplete sensor data as well as new observations that invalidate existing knowledge, and it can operate incrementally.

Roda and Musulin (2014) present a framework aimed at the extraction and representation of temporal abstractions from sensor data, primarily univariate time series. Temporal abstractions describe signal behaviour, such as a monotonically increasing trend over time. The representation of sensor data and temporal abstractions is achieved by means of ontology. The authors demonstrate how rule-based reasoning can be utilized to test whether sensor observations stated to be involved in a temporal abstraction are indeed within the time interval corresponding to the abstraction. The authors also demonstrate how rules can operate over temporal abstractions on multivariate sensor data to potentially detect a fault in an industrial plant.

The surveyed approaches for closing the 'semantic gap' are diverse along several dimensions. Some approaches envision the *automated* extraction of symbolic descriptions about an environment perceived using sensors (Gaglio et al., 2007; Negru, 2012; Ganz et al., 2014; Alirezaie and Loutfi, 2014). Automating the problem is extremely challenging, and we share Heintz et al.'s concern that it is only the simplest of autonomous systems that can be implemented using a single technique. Furthermore, the human is arguably never truly out of the loop. For instance, in rule-based systems, which can potentially operate autonomously on symbolic knowledge, experts are still required in the formulation of rules. Wavellite acknowledges the complexity of data processing and knowledge extraction in non-trivial applications, and explicitly includes experts in roles such as knowledge engineers, software engineers, and scientists.

Some surveyed architectures are layered similarly to Wavellite (Gaglio et al., 2007; Negru, 2012; Gorrepati et al., 2013). Some utilize semantic web technologies (Negru, 2012; Barnaghi et al., 2012; Janowicz, 2012; Gorrepati et al., 2013; Alirezaie and Loutfi, 2014) while other frameworks are based on XML (Conroy et al., 2011a,b). Semantic web technologies support the formal description of data semantics, and thus go beyond data structure. The resulting systems are superior in their support for knowledge reasoning, query, and discovery. The approaches have been evaluated in heterogeneous domains, some of which overlap with Wavellite applications, e.g. transportation systems (Liu and Zhao, 2005; Whitehouse et al., 2006) and scientific applications (Ganguly et al., 2007; Myers and Trevathan, 2013; Gorrepati et al., 2013). Compared to many surveyed architectures, Wavellite is meticulously evaluated on *several* non-trivial applications for problems in heterogeneous domains. Wavellite is tailored for the acquisition and representation of situational knowledge, i.e. the acquisition of assertional knowledge. In contrast, Ganz et al. (2014) and Janowicz et al. (2015) describe systems designed for the acquisition of terminological knowledge, i.e. designed to construct classes and relations. Systems also adopt various methods in knowledge acquisition, specifically similarity in metric spaces (Gaglio et al., 2007), data mining and machine learning (Ganz et al., 2014; Janowicz, 2012), rules (Gorrepati et al., 2013; Alirezaie and Loutfi, 2014), and techniques for explanation inference (Barnaghi et al., 2012; Henson et al., 2012; Ganz et al., 2014).

5.2 SITUATION ABSTRACTION

The concept of situation has been formalized in theories other than the situation theory at the core of this dissertation. Laying the foundations of the situation calculus, McCarthy and Hayes (1969) define situation as "the complete state of the universe at an instant of time." This definition is different from that of situation theory, where situations are *parts* of reality. Furthermore, McCarthy and Hayes localize a situation at a time instant. However, as in situation theory, McCarthy and Hayes (1969) argue that agents can only know facts about situations, i.e. agents cannot completely describe situations. The basic elements of the calculus are the set Sit of all situations, fluents, and actions. Fluents are functions whose domain is the set Sit. The fluent $snowing(p, s)$ asserts that it is snowing at place p in situation s. The range of a fluent can be $(true, false)$ or Sit. In the former case, the function is called a propositional fluent while fluents with range Sit are called situational. Actions change a situation to another (Worboys, 2005) and are performed by agents. McCarthy and Hayes model an action as part of the situational fluent $result(a, \alpha, s)$ having the situation as value that results when agent a carries out action α in situation s. According to McCarthy and Hayes, the known facts about a situation can be "used to deduce further facts about that situation, about future situations and about situations that [agents] can bring about from that situation." Levesque et al. (1998) propose a variant of the situation calculus that defines situations as histories, i.e. "finite sequences of primitive actions." This interpretation stands in contrast to McCarthy and Hayes, as well as situation theory, who understand situation as snapshots.

Gangemi and Mika (2003) offer an ontological analysis and formalization of situations. The proposed Description and Situations (D&S) ontology provides "a framework for representing [among other entities] situations at first-order, thus allowing a partial specification of [situations]." Central to D&S are state of affairs, descriptions, and situations. A state of affairs "is any non-empty set SoA of assertions $a_{1..n}$ that are individually coherent with the axioms in a first-order theory O," the ground ontology. A description "is an entity that partly represents a (possibly formalized) theory T (or one of its elements) that can be 'conceived' by an agent." Finally, a situation "is constituted by the entities and the relations among them that are mentioned in assertions $a_{1..n}$ from a SoA, and it is an entity in O that partly represents a (possibly formalized) model M for T, according to the axioms in O." The intuition is that "when a description is applied to a state of affairs, some structure (a 'situation') emerges." Following an example by Gangemi and Mika, a set of temperature values is a state of affairs, a climate change theory is a description, and a climate change history is a situation. Given a foundational ontology, such as DOLCE, D&S adds the two unary predicates D for descriptions and S for situations and a binary predicate `satisfies` holding between S and the set SD, subset of D, for situation descriptions. Notably, S

and D are 'social objects', i.e. objects that exist within social communication. In contrast, being structured parts of reality, situations in situation theory are arguably physical objects.

Systems in which situation is the key abstraction have also been proposed. Ye et al. (2012) review situation identification techniques in pervasive computing. The authors distinguish specification-based and learning-based approaches. Specification-based approaches "represent expert knowledge in logic rules and apply reasoning engines to infer proper situations from current sensor input." According to Ye et al., these approaches are particularly suitable for applications with "few sensors whose data are easy to interpret and the relationships between sensor data and situations are easy to establish." Learning-based approaches utilize "techniques in machine learning and data mining [...] to explore association relations between sensor data and situations." Ye et al. note that these approaches have the ability to analyse a large number of noisy sensor data, extract patterns, and deal with uncertainty.

For context-aware pervasive computing, Loke (2004) presents the notion of situation program. A situation program is a logic program, i.e. a set of rules. Given an entity, a known situation, and contextual information about the entity, the fundamental reasoning task is to determine whether the entity is in the situation. Entities can be persons, devices, or objects more generally. Contextual information about entities is obtained from sensors. Sensors are represented as sensor predicates, utilized in situation programs to query for contextual information about entities. A situation program is evaluated for a given entity by assuming "the situation and then explore its implications by forward chaining on the rules." In program evaluation, any constraints on sensor data are evaluated by querying the sensors and check the data against the constraints. If all constraints are satisfied, then the entity is recognized to be in the situation represented by the program.

Padovitz et al. (2004) present a reasoning engine that models context states and situation subspaces as geometries. A context state is a vector indexed in time consisting of context attribute values. A context attribute typically denotes a sensor. Thus, an attribute value denotes a sensor measurement value, indexed in time. A context state is a point, while a situation subspace is a region. A context state can thus be within a situation subspace. Given two situation subspaces, a context state can be closer (more similar) to one of them. As an example, consider the context state to be a vector with measurement values for heart rate, cadence, and stride length, and two situation subspaces for 'walking' and 'running'. Greater values reflect running activity and thus the context state vector falls within the 'running' situation subspace. The reasoning engine obtains raw data or "basic reasoned context" which are first analysed for low-level discrepancies (such as faulty measurement values) and then synthesized to basic situations. The engine attempts to discover and resolve conflicts in basic situa-

tions. Situations that are not discarded at this stage are finally composed into complex situations. Complex situations are the result of merging or intersecting situation subspaces.

For context-aware systems with persons as the entities in situations, and last-generation personal devices (e.g. smart phones and laptops) and online services (e.g. social networks and email services) as the sources from which context information used to characterise situations is extracted, Attard et al. (2013) present an adaptive situation recognition technique. Extracted context information is represented according to an ontology and matched against known situations using similarity metrics. A feedback loop involving users allows for annotating context information as positive or negative examples for recurring situations. Such annotation amounts to gradual training and the technique is thus adaptive.

Pongpaichet et al. (2013) present EventShop, a computing framework aimed at recognizing "evolving situations from massive web streams in real-time." As for other frameworks, at the core of EventShop is the challenge of bridging the semantic gap between high-level concepts and low-level data streams. In EventShop, the concept of situation is used as high-level abstraction and web data form the low-level data streams. The framework consists of two main components: data ingestor and stream processing engine. The data ingestor consumes data from the web and translates data to an internal unified format. The stream processing engine applies operators on translated data to detect situations. The relevant operators are determined by the situation recognition model formulated by domain experts. A situation recognition model is a query plan representing an ordered set of steps for processing data. Detected situations can be used to notify interested users. Pongpaichet et al. demonstrate EventShop on two use cases, flood alert and asthma relief. Dao et al. (2014) extend EventShop with functionality to process archived historical data, in addition to data streams.

The approaches discussed in this section share the aim of closing the 'semantic gap' with the approaches surveyed in the previous Section 5.1. However, the adoption of the concept of situation as the key abstraction is a distinguishing feature, one that is shared with Wavellite. By adopting both specification-based and learning-based methods, Wavellite implements a hybrid approach (Ye et al., 2012). Compared to the systems briefly discussed in this section, Wavellite also adopts semantic web technologies and, in particular, the STO to represent acquired situational knowledge.

5.3 EVENT ABSTRACTION

The ontological nature of the concept of event has been studied and formalized in the literature. Kowalski and Sergot (1986) introduce the event calculus, a classical logic formalization of time based on the notion of event, designed as framework for reasoning. Similarly to the situation calculus, the event calculus

models time-varying properties of the world as fluents. The basic statement of the calculus is that a fluent is true at some time point if it has been initiated earlier by an event and has not been terminated in the meantime by another event; otherwise it is false (Worboys, 2005; Miller and Shanahan, 2002). A number of alternative axiomatizations have been proposed in the literature and the event and situation calculi have also been compared (Miller and Shanahan, 2002).

Galton (2006) and Galton and Mizoguchi (2009) distinguish between durative and punctual events. The former take time to occur while the latter take no time to occur, and are thus instantaneous. Common to both event kinds is that an event has a beginning and an end, which act as anchoring points. Events have definite extension and are thus not open-ended; they are pieces of history. Galton also notes that an event that occurs over an interval "does not occur over any of the proper subintervals," i.e. events are non-dissective. Furthermore, events are not experienced directly. Rather, it is an event's constituent processes that are experienced, and events are described in terms of the constituent processes. Finally, as discrete chunks of history "events cannot meaningfully be said to undergo change." Discussing the ontological nature of events, processes, and objects Galton also presents some implications for a formal ontology, in particular for classification, and for logical representation and inference.

Worboys and Hornsby (2004) and Worboys (2005) propose the introduction of events in object-based geospatial models. The authors argue that the introduction of occurants, specifically events, in information systems and models is "needed to capture the mechanisms of change" in time-varying continuants, e.g. objects. In Worboys (2005), the author focuses on formal aspects of event specification and proposes an event-oriented model of the world in which "everything is event." In Worboys and Hornsby (2004), the authors adopt a hybrid approach with categories of entities other than events. Specifically, Worboys and Hornsby propose that geospatial events (and objects) are situated in a setting, which may be purely spatial (e.g. region), purely temporal (e.g. interval), or mixed spatio-temporal. An event (or object) "cannot be situated in more than one setting at the same time."

Authors have used the concept of event as the key abstraction in systems that process sensor data to acquire knowledge about the monitored environment. Although in situation theory events are situations in time, in existing systems event seems to stand in contrast to situation as the key abstraction and information object.

Building on their previous work, in which Devaraju and Kauppinen (2012) also highlight the "gap between low-level sensor observations and high-level descriptions about geographic events," Devaraju et al. (2014) develop an ontology for the representation of relations between geographic events and observations, and use the ontology "with a reasoning and querying mechanism to retrieve events and their sensing information." Geographic events are inferred

by means of rules expressed in terms of observed properties. Observed features are participants in events. The proposed ontology is deployed in a system that retrieves time series data, manages such data in an observational database, and infers information about events by means of ontology and rule-based reasoning. The system is evaluated for a use case in which blizzard events are inferred from weather observations. A blizzard is a geographic event and is described in terms of property-threshold constraints, e.g. visibility $\leq 1\,\text{km}$. Given a rule $p \rightarrow q$, with p a conjunction of constraints and q a concept assertion, it is possible to infer blizzard events. Given the ontological relation between geographic and observation events proposed by Devaraju et al., the system supports querying for the observation values related to blizzard events.

Authors have suggested to use Complex Event Processing (CEP) in systems aimed at detecting and representing events in sensor data. Taylor and Leidinger (2011) present a system that utilizes ontology to drive a user interface for event specification. The system translate such specifications into configurations for a CEP engine. The CEP engine processes (streamed) sensor data and generates events according to configurations, i.e. user-defined event specifications. Generated events can be utilized to alert users.

Llaves and Kuhn (2014) present a system that processes O&M observation data to CEP objects processed by a CEP engine. The system supports the registration of event patterns within the CEP engine. An event pattern is a pair consisting of a CEP statement (here encoded in Esper EPL) and an event type, i.e. a URI corresponding to an ontology class. Given an event generated by the CEP engine following a statement match, the system creates an individual, instance of the event type corresponding to the statement. The resulting RDF statements are persisted in an RDF database.

The concept of event and the use of ontologies to model event descriptions with knowledge obtained from data has been advocated for various domains. Wu (2012) present an ontology for the representation of tropical forest change events, such as deforestation. The proposed ontology extends the SSN ontology. The 'forest transition' observation is a central concept of the ontology, and is modelled as a subclass of the SSN observation class. The ontology aims at supporting the extraction of information about forest change events. For instance, given two forest transition observations over two years, if the observed deforested portion difference of a land unit is greater than zero, the land unit is inferred to be a participant in a deforestation event. Wu demonstrates how the extraction of such information about forest change events can be implemented using SPARQL.

Yu and Taylor (2013) present the Event Dashboard, an ontology driven user interface aimed at supporting users in the definition of event constraints over a sensor network, e.g. property constraints over observations such as air temperature greater than $15\,^{\circ}\text{C}$. Event Dashboard employs the SSN ontology. Specified

event constraints can serve in the definition of complex event queries. In contrast to systems that adopt rule languages, Event Dashboard encodes event constraints as ontology assertions. The authors demonstrate Event Dashboard for the specification of a high nitrogen constraint, which is relevant to algal bloom events. Building on Event Dashboard, Yu et al. (2014) propose an event detection system that processes sensor data to detect failure in pressure sewers.

Zhang et al. (2014) present a methodology for the detection and classification of anomalous events of salinity and turbidity in data collected from a water quality monitoring system. The methodology is distinctively data driven as it uses an adaptive trend model to detect anomalies in time series, agglomerative hierarchical clustering to group temporally close detected anomalies into events, and an additional clustering step to create groups of similar events. The resulting event clusters for salinity and turbidity are not semantically described, as it is common practice in ontology-based systems. The methodology would clearly benefit from machine interpretable descriptions of event clusters as it would enable a system to, e.g., create concept assertions for individual events detected in real-time monitoring. The methodology is relatively laborious as it consists of manual steps, whereas ontology-based systems are often argued to perform automated reasoning, once the ontology and the rules are properly specified. However, the methodology has also advantages over pure ontology-based approaches. The system can arguably support more complex event detection and classification tasks. Furthermore, as highlighted by the authors, the methodology requires little computational resources and can potentially be performed by the sensors, which can considerably reduce data transmission requirements. Ontology-based systems generally demand greater computational resources for their operation.

The approaches discussed in this section also share the aim of closing the 'semantic gap' with the approaches surveyed in the previous sections 5.1 and 5.2. However, the adoption of the concept of event as the key abstraction is a distinguishing feature. As we noted earlier, in situation theory events are situations in time, whereas scenes are situations in space. Therefore, it can be argued that systems which adopt the concept of situation as abstraction are also capable of representing events, similarly to the systems presented in this section. However, the literature seems to be inconclusive on whether events are distinct or subsumed by situations. Riker (1957) called situations "the boundaries of events" and events the action occurring between situations. This suggests that situations and events are entities of different character.

5.4 SITUATION AWARENESS

Systems designed to obtain, maintain, and improve situation awareness have been developed for various application domains. For emergency response and management, and building on previous work (Resch et al., 2007), Sagl et al.

(2012) present the architecture of an emergency management information system designed to acquire sensor data, integrate data into spatial decision support systems (e.g. GeoServer), apply geo-processing techniques (e.g. spatial interpolation) to obtain emergency information, and disseminate information to client devices that utilize map services (e.g. Google Earth) to visualize information. To maximise interoperability, the service-oriented architecture utilizes OGC standards, in particular the SOS for sensor data acquisition. A key factor in emergency response and management information systems is the time-critical decision-making process. Thus, sensor data must be acquired, processed, and disseminated to client applications in (near) real-time in order to support timely situation awareness. Based on the evaluation of their system, Sagl et al. highlight that expert feedback confirms that "up-to-date situational knowledge [...] significantly enhanced the decision-making process." The overall aim of such systems is thus to enable "live [situation] awareness of rescue forces and decision makers in emergency management."

Baumgartner et al. (2010) present BeAware!, a framework for ontology-driven information systems aimed at increased operator situation awareness. The authors present the framework within the road traffic management context. The framework obtains information from heterogeneous sources, and it is assumed that such information is represented according to some conceptual model, e.g. a domain ontology. The framework employs a domain independent situation awareness core ontology, utilized to integrate domain ontologies via alignment with the core ontology. Situation assessment in the framework is performed via instantiation of rule-based situation types. Hence, situation types are defined by means of rules, whereby the rule antecedent specifies the concept and relation types that must hold in order to instantiate the situation type specified by the rule consequent. In addition to enabling the integration of domain ontologies, the core ontology also facilitates the formulation of situation types.

Ontology-driven systems for situation awareness have been proposed also in the context of airport security. Tamea et al. (2014) sketch an architecture for systems that collect data from a sensor middleware and process the data to infer situations. The architecture includes standard OWL reasoning but the authors acknowledge that such reasoning "may not be sufficient to assess situations." Thus, the architecture allows for additional pre-processing or post-processing (programmatic) procedures. Tamea et al. propose a specialized ontology for the modelling of events and situations for airport security, which the authors compare to the domain independent STO. The authors evaluate the performance of creating new events using the two ontologies and conclude that their ontology for airport security has a better performance. This is attributed to the more lightweight structure. However, the authors seem to mistake the infon as the type of events while, in situation theory, events are more accurately a type of situations.

Furno et al. (2011) describe an ontology-based situation awareness architecture with three main components for perception, comprehension, and projection, following thus the three level model by Endsley. The architecture is discussed in the context of ambient intelligence, specifically for the problem of face detection. The authors underscore the importance of uncertainty management, and use the Fuzzy Situation Theory Ontology (Furno et al., 2010, FSTO) to model the uncertainty of infons and situations. In FSTO the polarity states the *degree* to which infon objects stand in the relation. Its aim is to support the approximate evaluation of situations.

In addition to land, air, or maritime situation awareness, environmental monitoring and data analysis are relevant also to space situation awareness. Perron (2014) discusses space *weather* situation awareness, in particular. Space weather is primarily driven by the Sun and "encompasses several components of the Sun-Earth system, such as the variable solar wind, sunspots, solar flares, coronal mass ejections, interactions with the Earth's magnetosphere and ionosphere, and the production of the aurora." Perron discusses space weather situation awareness within the military context. However, space weather events can negatively affect technology and humans, more generally. Hardware and software systems play a key role in space weather situation awareness, including in the prediction of space weather events.

Salmon et al. (2012) review road transport-related situation awareness research applications. The authors note that situation awareness "has received far less attention in a road transport context," compared to domains including aviation, military, air traffic control, rail, process control, and healthcare—even though it is "highly applicable" and incontestable that situation awareness "is required for different road user tasks." However, the authors argue that the individualistic driver centric approach to situation awareness is inadequate because it overlooks important components other than the individual driver, "such as pedestrians, motorcyclists, cyclists and infrastructure." The authors thus advocate for a systems approach to situation awareness in road transport.

Hasan et al. (2011) propose to enrich sensor data using semantic web (linked data (Bizer et al., 2011)) technologies and utilize CEP as a means to achieve situation awareness. Situations are "expressed in the configuration of the CEP engine in the form of an event pattern." Enrichment of sensor data is dynamic, i.e. determined at run-time, and includes translation from, e.g., O&M and SensorML XML to RDF, and utilizes a semantic similarity measure between data and defined situation patterns to guide enrichment. For instance, given data by an energy usage sensor and a situation pattern for energy consumption on a floor, an RDF statement relating the device to the floor on which it is installed is an enrichment guided by semantic similarity. This is because both the RDF statement and the situation pattern involve the term 'floor'. Candidate RDF statements are obtained by dereferencing linked data. The method is briefly evaluated for an energy management use case.

Jakobson et al. (2006) present a framework for situation management. According to the authors, situation management is a process concerned with sensing and information collection, perceiving and recognizing situations, analysing past and predict future situations, as well as reasoning, planning and implementing actions. Situation management is a goal-directed process. Jakobson et al. discuss situation modelling and provide a definition for situation, noting that "situation modelling is still in the process of establishment, where many notions, including the notion of a situation itself, are still the subjects of ongoing debate and study." The authors cite "the need for an integrated management of complex dynamic systems" to be an important driving force behind the advancement of situation management.

Salfinger et al. (2014) highlight that so far there has been little focus on "supporting the different phases of knowledge management in [situation awareness] systems, which encompasses the acquisition, representation, validation, maintenance and reuse of knowledge gathered for and during the use of these systems." Salfinger et al. propose a tool suite for the management of situational knowledge in situation awareness systems. The authors discuss the relevant tasks including knowledge acquisition, representation, validation, adaption, exploration, and exploitation. Salfinger et al. plan to evaluate the suite for traffic management.

For maritime situation awareness, Velikova et al. (2014) present a system capable of collecting, and reasoning on, data and information to provide human operators with actionable knowledge, in real-time. The system uses heterogeneous data and information sources, e.g. automatic identification system transmitters installed on ships and commercial ship databases, to assess the truthfulness of identification information transmitted by ships, and monitor ship behaviour in order to detect illegal or dangerous activities. Velikova et al. underscore that "early [situation awareness] systems were only capable of collecting low-level sensor data [...] to generate ship tracks," leaving the recognition of abnormal patterns to operators, whereas newer systems are increasingly capable of identifying abnormalities automatically. A consequence of this "shift from 'mental reasoning' to 'automated reasoning' [...] is [operator] relief from the burden of dealing with all ships to dealing with only those ships that really matter."

As noted by Salmon et al. (2012), situation awareness has traditionally received attention in domains such as aviation, military, and air traffic control. Far less attention has been given to situation awareness in intelligent transportation systems. Paper **II** is a contribution to this domain. Furthermore, we argue that situation awareness has been applied even less to environmental monitoring in scientific applications, even though it is arguably applicable. Traditionally, situation awareness systems include one or more operators, i.e. a person who typically operates machinery, such as aeroplanes or ships. We build on this work and argue that in situation-aware environmental monitoring systems for research it is scientists that can be the operators, receivers of knowledge about situations

involving environmental phenomena, such as new particle formation, acquired from environmental sensor network data processed by means of computational models. Thus, the most important distinguishing feature between the systems reviewed in this section, and arguably situation awareness systems more generally, and the systems in Wavellite applications discussed in Chapter 4 are the operator and the elements in the environment for which situation awareness is obtained and maintained.

5.5 DATA MANAGEMENT

In addition to the problem of knowledge extraction from processed data acquired from environmental sensor networks, and the representation of acquired knowledge, the Wavellite framework, and thus the applications presented in Chapter 4, also faces the problem of data management, both raw sensor data and processed datasets. This subtask is motivated at least by the facts that "[e]nabling machine interpretability [...] requires a common approach to organise and structure the data" (Barnaghi et al., 2009) and that "integrating [data] into forms usable for environmental analysis and modelling can be highly time-consuming and challenging" (Hill et al., 2011).

Given the rapidly increasing use of sensors in various domains, and the consequently growing amount of data generated by such devices, it is unsurprising that sensor data management, too, is a research topic of continued and increasing importance. Balazinska et al. (2007) note that "we have placed too much attention on the networking of distributed sensing and too little on tools to manage, analyse, and understand the data." While useful as a starting point, Balazinska et al. underscore that conventional database management systems "have several critical shortcomings that prevent using them directly to process live sensor data." The authors discuss data management challenges of the 'worldwide sensor web' vision and present related recent advances in research. Challenges include the quality of collected data, which may be uncalibrated, missing, faulty, and does generally not fit a uniform space-time grid; the management of temporal and spatial data, which often requires programming skills to perform calculations, visualization, or statistical analysis; the complexity of data analysis workflows; the lack of integration between conventional data management systems and modelling software; data uncertainty and provenance; distributed streamed data and query processing.

Marascu et al. (2014) present TRISTAN, a "data management system for efficient storage and real-time processing of fine-grained time series data." The system persists only the most informative parts of time series in an optimized compressed representation and performs queries directly on the compressed representation, avoiding thus having to decompress the data. As a result, TRISTAN

achieves high compression ratios and query performance improvement within one or two orders of magnitude.

Chang et al. (2006) present a "solution to data storage and management that provides a uniform and consistent method for publishing and sharing [...] sensor network data." The system supports sending and retrieving data via standardized formats. Queries can be formulated to include search on geographic location, sensor type, time intervals. The primary aim of the system is to hide the complexity of retrieving and processing data of sensor networks with heterogeneous data storage and management mechanisms.

Addressing the challenges of accessing and integrating environmental sensor network data for analysis and modelling, Hill et al. (2011) present a web-based virtual sensor system that "creates real-time customized data streams from raw sensor data," using spatio-temporal and thematic transformations, and supports "the publication of both the derived data products and the workflow [and provenance] that created them."

Horsburgh et al. (2011) describe the "architecture and functional requirements for an environmental observatory information system that supports collection, organization, storage, analysis, and publication of hydrologic observations." The architecture is designed to support time series data collected from stationary sensors and is evaluated for a river in Utah. The supported functionality is closely aligned to ENVRI-RM functionality. The architecture includes data analysis, which Horsburgh et al. describe as "the process by which data are inspected, modelled, and visualized with the goal of increasing understanding of hydrologic processes." In the architecture, it is the scientist who searches and retrieves relevant data from the information system, an exploratory task as "[u]sers don't always know exactly what they are looking for." The scientist also performs data analysis and interpretation. This can be understood as knowledge extraction from data, and it typically involves software, e.g. for statistical computing, but is otherwise performed manually. More importantly, the architecture seems to not allow for feedback of knowledge obtained from data analysis and interpretation into the environmental observatory information system.

Dow et al. (2014) provide a survey for the best practices in water data management and retrieval, sharing, and visualization in informatics platforms. Their focus is on user and researcher "ability to access and analyse the data more effectively." The authors discuss three core informatics functions. The first function is to provide data of value, meaning that the content is comprehensive and "in sufficient quantities to derive statistically meaningful conclusions," and the "associated metadata contains enough detail for end users to assess reliability and compatibility," including provenance information. The second function is to support the exchange of data, in order to make useful data accessible using web services, machine-readable data conforming to standardized XML, and ontological solutions for machine-interpretable environmental data interoperability. The

third function is to help users analyse the data, which includes functionality for data search and discovery as well as data visualization. The review seems to suggest that state of the art water informatics platforms are (so far) not concerned with integrating models for knowledge extraction. With the exception of trivial alerts based on user-defined thresholds for parameters of interest, information and knowledge extraction is generally left to the user, e.g. researcher. Moreover, there seems to be no feedback loop for, manually or automatically, extracted knowledge into informatics platforms which thus do not manage or process knowledge.

The semantic description of sensors and their data using ontologies has received considerable attention in the literature. Sheth et al. (2008) discuss the semantic sensor web "in which sensor data [are] annotated with semantic metadata to increase interoperability as well as provide contextual information essential to situational knowledge." Sheth et al. specify that such annotation involves, in particular, spatial, temporal, and thematic semantic metadata. The aim is "to provide enhanced descriptions and meaning to sensor data."

The semantic sensor web extends the sensor web, which "refers to web accessible sensor networks and archived sensor data that can be discovered and accessed using standard protocols and application program interfaces" (Botts et al., 2007, 2008). Toward the sensor web vision, the Open Geospatial Consortium (OGC) develops "a suite of specifications related to sensors, sensor data models, and sensor web services" (Sheth et al., 2008). However, these XML-based specifications are designed for syntactic interoperability between information systems (Egenhofer, 2002); they cannot achieve semantic interoperability.

Addressing this limitation, Probst (2006) suggests to align key terms of OGC O&M to the DOLCE foundational ontology (Masolo et al., 2002). With the semantic sensor web, Sheth et al. (2008) extend the syntactic XML-based metadata standards of the OGC with OWL-based semantic metadata standards of the W3C. Sheth et al. propose a mechanism whereby semantics are added into XML documents by annotating (OGC) XML with terms defined in ontologies. The authors demonstrate the mechanism by annotating a timestamp encoded in O&M with the term OWL-Time `Instant`.

Ontologies were soon developed. Compton et al. (2009) provide a survey of early efforts in the semantic specification of sensors. Some ontologies are designed to primarily support the description of sensor types, others focus on sensor data, and some support the description of sensor systems, their components, structure, and processes. Today, the most notable result in the semantic specification of sensors is arguably the SSN ontology.

The adoption of semantic web technologies for sensor data management has been advocated. Lewis et al. (2006) argue that "semantics can enhance data management in sensor networks." The system presented by Lewis et al. manages sensor data in (daily) RDF files, a practice that, in light of state of the art RDF databases, is arguably antiquate. However, the authors underscore the ability

of the RDF (graph) data model to represent semantic associations between data, and the possibility of using such relations in the formulation of queries.

Le-Phuoc et al. (2011) present a Linked Stream Middleware for the collection of heterogeneous sensor data, their translation into RDF data that conforms with the SSN ontology, and the access of RDF data. The middleware architecture consists of four layers. At the bottom, the "data acquisition layer provides wrappers to collect sensor readings and transform them [to RDF]." The linked data layer receives RDF data from the data acquisition layer and supports further annotation and enrichment of sensor data in RDF by linking to external RDF data sources. The data access layer supports declarative querying of RDF data enriched by the linked data layer. Finally, at the top, the application layer is concerned with application development. The middleware is designed to support the collection and transformation of large volume data. The authors claim that their instance was handling over 100,000 data sources. However, the middleware does not address knowledge extraction from data and knowledge representation.

Wang et al. (2011) describe a "semantic technology-based approach to ecological and environmental monitoring." The authors develop an upper ontology for monitoring, and deploy the approach in a system that "integrates environmental monitoring and regulation data from multiple sources" using semantic web technologies. They also argue that environmental monitoring systems must at least model background environmental knowledge, observational data items collected by sensors and humans, and environmental regulations. The third type of domain knowledge, i.e. environmental regulations, is arguably not a necessary part of environmental monitoring systems in arbitrary applications. For instance, Paper **III** presents an application in which aerosol scientists are not concerned with the 'regulation' of new particle formation.

Lefort et al. (2012) use the SSN ontology and the QB vocabulary to publish temperature data released by the Australian Bureau of Meteorology as tabular time series (tab-delimited data files) in RDF as a 'Linked Sensor Data Cube'.

Ahmedi et al. (2013) present an ontology for water quality management. The proposed ontology was "developed to support water quality classification based on different regulation authorities." The ontology is based on the SSN ontology and supports the modelling of sensor data, regulations published by authorities, sources of pollution, and expert knowledge about the water (quality) domain, in particular rules.

Abecker et al. (2014) present a sensor and semantic data warehouse "able to store and provide sensor, measurement and forecasting data, as well as semantic knowledge about the water-supply chain." The software architecture separates sensor and semantic data into distinct stores. Specifically, sensor data in form of OGC WaterML 2.0 (Taylor, 2014) is managed by a conventional relational database management system whereas data with irregular and complex relationships is managed by a knowledge base. According to Abecker et al.,

to manage the sensor data with the RDF database "seemed not feasible and promising." However, a drawback of the approach is the resulting 'technology gap' which means that it is not possible to evaluate sensor data and semantic data in a single query.

The semantic annotation of sensor data is a subtask of sensor data management with semantic technologies, and is frequently discussed in the literature. Studied for a building fire emergency scenario, Huang and Javed (2008) present an architecture for a system that enriches sensor data with semantic information such that the data can be understood and processed by applications with different purposes.

Discussed for the transportation domain, Stewart Hornsby and King (2008) demonstrate an approach for linking data about vehicles observed by a sensor network, and managed by a database, with ontology classes. The work builds on aligning database schema with ontologies, and proposes to link database instance data with ontologies.

Wei and Barnaghi (2009) propose to annotate sensor data with concepts of existing knowledge bases, such as DBpedia (Auer et al., 2007), following the linked data principle, and utilize semantic reasoning over sensor data to infer new knowledge and answer complex user queries. The authors argue that linking sensor data with qualitative annotations, e.g. 'cold' for $-15\,^{\circ}\mathrm{C}$ measured ambient air temperature, increases the value and usefulness of the data.

Müller et al. (2013) present an approach for transforming JSON formatted sensor data to RDF using mappings between JSON document elements and ontology concepts and properties. For this purpose, the authors develop a mapping language, which is used to create transformation scripts. The script language also supports the invocation of functions designed to retrieve relevant data from external sources. The feature can be used to further enrich RDF data with relevant data that is not available in the original JSON data. The example provided by Müller et al. is for a function that takes coordinate data available in JSON and uses the Google Geocoding API to obtain an address. In addition to coordinate data, the resulting RDF data will thus be enriched with address data.

Moraru and Mladenić (2012) present a framework for the semantic enrichment of sensor data. The purpose of the framework is to support the automated translation of sensor data to, and the querying of sensor data in, RDF. However, the proposed framework is conceptual and seems to lack of an implementation.

Calbimonte et al. (2012) propose to learn semantic properties of observations from sensor data. The approach uses linear approximations of time series slopes and a similarity-based classification of slope distributions to learn the property type observed in measurement.

Another subtask of sensor data management is data access. In order to abstract from the heterogeneity of devices in sensor networks, service oriented principles have been adopted to model sensors as services and thus enable ac-

cess to sensor data through standard service technologies. With semantic sensor service networks, Wang et al. (2012) propose a generic framework that models sensors as services and supports the semantic description, seamless service-oriented connectivity, discovery and composition of sensor services. Of concern to sensor services are, among other issues, semantic registries for sensor metadata (Chaves et al., 2013) and the matching of sensor characteristics and service requirements for correct integration (Bröring et al., 2012). Some of the OGC service specifications have been extended with semantic features. An example is the semantically enabled SOS proposed by Henson et al. (2009).

Wavellite handles data management at the observation, derivation, persistence, and access layers. It builds on semantic web technologies. Wavellite is thus similar to the SSN ontology based systems presented in this section, and in contrast to more traditional systems that rely on relational database management systems and XML technologies for data interchange. However, the main distinguishing feature between Wavellite and other sensor data management systems is the acquisition of situational knowledge from data, and the representation and processing of knowledge. Especially in environmental science, informatics platforms for data management tend to support classical ENVRI-RM functionality, and are generally intended to be used by researchers as repositories for data that serve analysis, and thus information and knowledge extraction. However, these informatics platforms seem to not support the feedback of acquired knowledge into the platform, and thus the management and processing of knowledge. In contrast to informatics platforms for data management—which may have good support for handling data, including import and export of data in various formats—the Wavellite data management layers primarily serve the purpose of situational knowledge acquisition. Hence, the functionality supported by these layers is limited to those strictly required for knowledge acquisition in various applications.

5.6 RELATED AREAS

We discuss related techniques in environmental monitoring, specifically remote sensing; techniques in data management and processing, such as stream processing and data fusion; methods in data processing, data analysis and mining, and machine learning on sensor data; and other domains in which obtaining symbolic descriptions about an environment perceived using sensors is a research problem, such as robotics, context awareness, ambient intelligence.

Sensor networks have found diverse application in environmental monitoring. Mainwaring et al. (2002) describe the core components of a sensor network architecture for the domain of habitat monitoring, of which the authors present an instance for monitoring seabird nesting environment and behaviour. The authors are primarily concerned with hardware infrastructure, and related re-

quirements, and mention that sensor data are managed by PostgreSQL. Moumen et al. (2014) present a platform aimed at real-time groundwater monitoring. The platform builds on OGC services and technologies, such as OGC SOS and the 52° North SOS server, and is evaluated for a region in Morocco.

In our work, we have generally assumed that monitoring occurs *in situ* by means of environmental sensor networks. Remote sensing, in particular satellite based techniques, is an alternative mode of environmental monitoring and comes with the primary advantage of large scale spatial coverage. To highlight a few studies, Ackland et al. (2012) use "remotely sensed data for the purpose of flood monitoring in terms of measuring flood extent and estimating flood volume at continental scales." Using remote sensing data, Xiao et al. (2014) present an OGC-based system for dust storm detection and visualization.

Due to characteristics such as heterogeneity, volume, frequency, and noise, data streams are notoriously challenging to process. Over the past decade, various stream processing and querying engines have been proposed (Babu and Widom, 2001; Yao and Gehrke, 2002; Chandrasekaran et al., 2003; Madden et al., 2005; Abadi et al., 2005). The use of CEP techniques to process data streams has been advocated (Bonino and Corno, 2012).

Gaber et al. (2005) provide a review of data stream analysis and mining, related systems and techniques, and research challenges. Data stream mining is concerned with extracting knowledge from data streams. Techniques in data stream mining are thus arguably of interest to near real-time knowledge extraction from environmental sensor network data. The authors briefly review clustering, classification, frequency counting, and time series analysis techniques. Clustering and classification, in particular, are broad technique categories originally developed for static datasets. The static and the streamed contexts have different demands on algorithms because in the latter case the view on data is limited to a window, and algorithms thus operate incrementally. As for research challenges, Gaber et al. note, among other, the poorly supported handling of continuous data streams by traditional database management systems; the design of energy and memory efficient techniques that can operate on resource constrained devices; the representation of data mining results and their transfer over limited bandwidth communication links; and the visualization of results on mobile devices.

Calbimonte et al. (2010) discuss the ontology-based access to data streams and present a SPARQL extension for streaming data that supports operators over RDF streams. An RDF stream is a sequence of pairs consisting of an RDF triple and a timestamp. Similar extensions to SPARQL have been proposed in the literature (Bolles et al., 2008; Barbieri et al., 2009). The efficient transmission of RDF streams using data compression techniques has also been addressed (Fernández et al., 2014). Stream reasoning, i.e. logical reasoning on data streams, has also drawn interest recently (Della Valle et al., 2009; Barbieri et al., 2010;

Margara et al., 2014) and Wetz et al. (2014) discuss the integration of RDF streams in environmental information systems, noting that the blending of static data sources and dynamic data streams "is non-trivial and major advances still need to be made in this area."

Historically developed primarily for military applications, sensor data fusion has seen also non-military applications, e.g. in machinery condition maintenance and robotics, and has also tackled some of the problems underlying this dissertation. Hall and Llinas (1997) describe how the detection of a target in sensor data, e.g. an aircraft; the determination of its properties, such as position and velocity; the estimation of target identity, e.g. F-16 aircraft; and the interpretation of the target's intent, require methods in signal processing, pattern recognition, and knowledge-based methods. Such a process is clearly one of situation assessment, and situation awareness models the perception, comprehension, and projection of the aircraft. In this context, the fusion of sensor data "may increase the accuracy with which a quantity can be observed and characterized." This constitutes an obvious benefit for tactical military systems. However, Hall and Llinas acknowledge that the "actual implementation of effective data fusion systems is far from simple." The authors list environmental monitoring as one of the non-military applications of sensor data fusion and highlight the use of remote sensing image-based techniques to monitor weather and natural disaster.

Methods in data processing, data analysis and mining, and machine learning, in particular methods applied to sensor data, are important to Wavellite applications. The framework borrows such methods from the literature, and implementations from existing software packages. Much work exists in the literature in which authors have applied methods in data processing, analysis, mining, and learning on sensor data to extract information.

Mahmood et al. (2013) review data mining techniques for wireless sensor networks. The authors highlight how some of the inherent characteristics of wireless sensor networks pose challenges to traditional data mining workflows. According to the authors, "traditional data mining is centralized, computationally expensive, and focused on disk-resident transactional data." In contrast, sensors are constrained in battery lifetime, memory, communication, and computational resources; sensor networks are often sources of large amounts of fast sampled data; sensor data are located in space-time; sensor networks are distributed systems with potentially dynamic topologies; applications may require data mining to operate in real-time and models need to be updated as the underlying phenomenon changes over time. Mahmood et al. discuss methods in frequent pattern mining, utilized to discover groups of variables that co-occur frequently in datasets; sequential pattern mining, utilized to discover frequent subsequences in sequence databases such as time series; clustering, utilized to group data so that the similarity of data in the same cluster is greater than the similarity of data in different clusters; and classification, utilized to assign classes to data objects.

Fu (2011) and Esling and Agon (2012) provide reviews for time series data mining. According to Fu, fundamental tasks in time series data mining include representation, utilized to reduce the size of time series while retaining the fundamental shape characteristics; indexing, required to efficiently retrieve time series; the computation of the similarity between time series or time series subsequences; time series segmentation, utilized in preprocessing, trend analysis, or discretization; time series visualization, utilized for user analysis; and mining tasks such as pattern discovery, clustering, classification, rule discovery, and summarization. Esling and Agon include query by content, anomaly detection, and prediction as further tasks. Liao (2005) surveys clustering of time series data, in particular. Similar techniques have also been proposed for spatial and spatio-temporal data mining (Koperski et al., 1996; Roddick et al., 2001).

Lane and Georgiev (2015) utilize neural networks on mobile devices to map (processed) sensor data to context, for tasks such as activity recognition. Compared to desktop, server, or cloud computer systems, mobile devices operate under various constraints, such as limited battery lifetime as well as constrained processing and memory resources. Performing machine learning tasks directly on mobile devices may require innovative algorithms. While the computational resources available to mobile devices such as last-generation smart phones are often more advanced than those available to sensors used in environmental monitoring, research that aims at pushing advanced processing and analysis algorithms into sensing devices with constrained computational and energy resources is arguably relevant to sensor network applications, as devices could perform advanced computational tasks lower in the infrastructure and reduce the amount of data communicated to components at higher levels of the architecture. However, in scientific applications research communities are arguably keen on retaining (raw) sensor data, as it may serve future research goals that are unforeseen at the time of data acquisition.

Data mining methods have also been applied to data on the semantic web. Rettinger et al. (2012) survey statistical approaches to mining the semantic web. In contrast to deductive reasoning typically supported in knowledge-based systems, inductive methods better handle very large, noisy, inconsistent, uncertain and missing data. Rettinger et al. underscore that "machine learning has been mostly considered as a tool to enrich or extend ontologies on the schema level," i.e. to learn terminological axioms. The authors thus focus on using data mining methods at *instance* level to learn assertions. Relevant tasks in this context include the prediction of class membership and property values of instances, the prediction of relations between individuals, as well as instance clustering and relation classification.

Some authors argue that large quantities of data and methods in data mining and machine learning provide "opportunities for the greatest scientific and technological advances of the early 21st Century" (Peters et al., 2014). While

research has been addressing the collection, curation, access, and processing of large quantities of data with hardware and software infrastructure, Peters et al. underscore that "big data are not readily accepted or utilized by most ecologists as an integral part of their research because the traditional scientific method is not scalable to large, complex datasets." The authors argue that "what is needed is a knowledge-driven, open access system that 'learns' and becomes more efficient and easier to use as streams of data, and the number and types of user interactions, increase." Peters et al. sketch the architecture of such a system, called Knowledge Learning and Analysis System. The key feature of the system is arguably the integration of hypothesis-driven and data-intensive machine learning scientific approaches. Interesting here is also the remark that "the current focus [in environmental research infrastructure] is on open access source data and metadata." Peters et al. argue that "a more efficient use of resources will occur if the derived data products and analyses are also in the public domain and continually modified as more scientists use and learn from the data." A claim of this dissertation is that not only derived data products should be accessible: also knowledge products, i.e. the result of knowledge acquisition—possibly achieved by means of machine learning methods—can, and perhaps should, be curated by environmental research infrastructure, and thus be accessible.

The idea of representing knowledge derived from (geographic) datasets dates back to at least Mennis and Peuquet (2003). The authors suggest that "geographic data models that support knowledge discovery must represent both observational data and derived knowledge." Mennis and Peuquet stress the importance of incorporating aspects of knowledge representation into the knowledge discovery in databases process. The authors discuss their approach for a case study in which expert knowledge is used to create a typology of storm types and their properties, represented in a database context. Storm individuals of a given type are then extracted from meteorological (observational) data.

The problem of obtaining symbolic descriptions about an environment perceived using sensors is common to several other domains. In video surveillance, an important task is to automatically understand events occurring in scenes monitored using video surveillance sensor networks. Of particular interest are densely populated environments, e.g. cities, airports, subways. Doulaverakis et al. (2011) stress that "manual observation of multiple camera feeds is not possible" and argue that "high-level intelligent reasoning for event inference" are thus critical features. Applied to security surveillance environments, Doulaverakis et al. present a software architecture for a sensor information fusion system. The architecture consists of four layers and includes monitoring with sensors, signal processing of sensor data, and mapping of data to ontology concepts. The authors use the STO and thus the concept of situation as the core abstraction for information about events observed in surveilled environments.

According to Fernández et al. (2013), video surveillance sensor networks suffer from "an overabundance and overflow of data which does not directly translate into information." The emphasis is on directly, i.e. the data are captured but systems do not process data to information in (near) real-time. As a result, video surveillance data are predominantly used to understand scenes in the past, e.g. to obtain evidence for a past crime, rather than for prevention. Fernández et al. present the architecture of a surveillance platform. The employed visual sensors natively detect movement in the monitored scene and transmit object motion data as XML. XML data are then processed to detect 'irregular activity'. The system searches irregular activity by learning recurrent patterns in motion, by semantic characterization (i.e. interpretation) of motion in the scene, and by means of user-defined rules, which may include atoms that implement complex functions on fused data from heterogeneous sources. Irregular activity is detected if patterns are different from recurrent patterns or if a semantic reasoner or a rule infer irregular activity. Upon detected irregular activity, an alarm message is dispatched to first responders, e.g. the police, who can then request the system for a live video stream on the scene.

Researchers in robotic systems have studied the problem of providing robots with a symbolic representation of its environment perceived using heterogeneous sensors. In a robotics context, Coradeschi and Saffiotti (2000) discuss the problem of anchoring symbols to sensor data. According to the authors, "[a]nchoring is the process of creating and maintaining the correspondence between symbols and percepts that refer to the same physical objects." Coradeschi and Saffiotti (2003) also provide a definition for anchoring that substitutes 'percepts' with 'sensor data'. Anchoring is thus the correspondence between symbols and sensor data. Coradeschi and Saffiotti argue that, provided with a symbol system utilized to reason about abstract knowledge, in order to execute tasks a robot must be able to anchor the symbols for abstract knowledge to the data generated by its sensors. The authors thus consider anchoring "a necessary component of any physically embedded symbolic system." Curiously, Coradeschi and Saffiotti borrowed the term 'anchor' from Barwise and Perry's situation semantics.

Coradeschi and Saffiotti (2003) discuss various challenges of anchoring. Of particular interest to environmental monitoring is the challenge of uncertainty and ambiguity. The authors highlight three aspects. First, "[s]ymbolic properties often do not have a precise definition in terms of measurable attributes." Second, there can be a "mismatch between what we would like to discriminate at the symbolic level [...] and what can be actually discriminated by the sensors." Third, "at the symbolic level we can refer to objects with a specific identity [...] while the perceptual system is not in general able to perceive the identity of an object but only some of its properties."

Loutfi et al. (2008) discuss the inclusion of knowledge representation and reasoning in perceptual anchoring. Objects anchored in data are described in symbolic form in a knowledge base using appropriate concept assertions, and role assertions to represent object properties. The authors conclude that such integration can facilitate a natural and effective interaction between people and robots, which is demonstrated in the context of an intelligent home environment. The use of semantic web technologies for robot knowledge representation has also been advocated (Gurău and Nüchter, 2013; Persson et al., 2013; D'Este et al., 2014).

For robot perception, Shanahan (2005) present a "formal abductive account of the means by which low-level sensor data [are] transformed into meaningful representation." Given a background theory (background knowledge) and 'observation sentences', abduction attempts to provide an explanation for the observation sentences. Explanation is reasoning from effects to causes, whereby observation sentences are the effects and physical phenomena are the causes. Abduction has been used by various authors, in particular also for perception in the semantic sensor web (Henson et al., 2012; Barnaghi et al., 2012; Ganz et al., 2013). Shanahan underscores that, in general, the observation sentences are not sensor data in its raw state but refined descriptions, such as object edges extracted from an image, possibly represented as logic formulas. In practice, such refined descriptions may not be readily available to a system. Systems must thus also address the problem of how to obtain refined descriptions. Furthermore, the reasoning task in applications may be from causes to effects, i.e. prediction, meaning that successful systems are required to support various reasoning modes.

Maurelli et al. (2014) present a system, intended for marine cognitive robots, that processes sensor data, identifies features such as lines and circles, and accordingly populates a knowledge base. Information for basic features are then processed to information about more complex concepts, in particular underwater structures. The authors underscore that the use of robots in practice continues to be hindered by "their limited ability to cope with unexpected events and environments." Maurelli et al. argue that knowledge representation can at least partially address this problem by enabling robots "to model and reason with the uncertainty in the world," such as unexpected obstacles along survey paths of autonomous underwater vehicles.

Anchoring is a concrete aspect of symbol grounding (Harnad, 1990), which is the problem of "how to give an interpretation to a formal symbol system that is based on something that [...] is not just another symbol system" (Coradeschi and Saffiotti, 2000). Symbol grounding is a more general problem than anchoring. Cregan (2007) adapts the symbol grounding problem to the semantic web, and notes that ontological entities have formal semantics but "lack a pragmatic semantics linking them in a systematic and unambiguous way to the real world

entities they represent." Cregan argues that this constitutes a problem for the semantic web in that it is possible for a symbolic system, such as the semantic web or an ontology, to produce logically valid inferences which, however, have no meaningful correspondence in the real world.

For information systems that utilize sensor data, Fiorini et al. (2013) adapt the symbol grounding problem to the problem of grounding the symbols of a domain ontology. Fiorini et al. study how to represent the grounding link between sensor data and domain ontologies and argue for the explicit representation of this link. The proposed symbol grounding framework consists of three levels: domain, quality, and signal. The domain level contains domain knowledge, terms including concepts and qualities such as apple and colour. The quality level uses attribute functions to specify the relations between qualities, concepts, and quality values, for instance an attribute function that specifies the quality of having a certain colour, such as the *red* colour of kinds of red apples. Finally, the signal level uses symbol detector functions to link quality values to patterns in raw data. Symbol detectors search data for the existence of qualities. The framework can induce the existence of an individual of a particular concept if enough qualities of the concept are found. Fiorini et al. clarify that the proposed framework does not specify how symbol detectors are implemented in practice and note that "they can be implemented as simple logic rules, as well as more complex software systems, involving signal processing methods." In the application presented by the authors, "the link between the quality level and the signal level is hard-coded" and symbol detectors are implemented as programming libraries.

With shared aims and problems, context-awareness is another research area related to this dissertation. Dey (2001) defined context as "any information that can be used to characterise the situation of an entity." Dey specifies that "[a]n entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and application themselves." Dey further states that "[a] system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task." According to Dey, a description of the states of relevant entities is a situation abstraction.

Sensors can be a source of context and methods have been developed to extract context from sensor data. Castelli et al. (2009) propose the W4 context model, whereby a fact that occurs in the world is expressed as four-fields tuple consisting of data or information for Who, What, Where, When. The authors contend that "most information about the world can be [...] represented in terms of [the] four 'W's." According to Castelli et al., sensors are one possible source of data or information relevant to four-fields tuples of the W4 context model. The authors envision the possibility of using ontology to represent four-fields tuples such that the W4 model is "usable in open and dynamic scenarios."

Kessler et al. (2009) evaluate the use of semantic rules for context-aware geographical information retrieval. The authors remark that choosing a suitable location, e.g. for sport activities, generally depends on various factors, such as environmental conditions, personal skills and preferences, or social aspects. Information for such factors can be used to characterise the situation of a user and is thus context. Context information may be numeric data acquired from sensors, such as a thermometer. Kessler et al. then propose to utilize SWRL "for personalized mappings between the numeric sensor world and information stored in ontologies." In particular, the authors contend that SWRL built-ins play a central role in the mapping, and discuss the design of a built-in that retrieves the value of an observed property from a SOS.

Data of sensors typically included in mobile devices, such as accelerometers, have been used to infer behaviour and context. Lane and Georgiev (2015) highlight that to reliably infer "user behaviour and context from noisy and complex sensor data collected under mobile device constraints remains an open problem." Cricri et al. (2014) use the data of auxiliary sensors typically installed in camera-enabled electronic devices to obtain contextual information about user generated videos. Using a single camera, the authors demonstrate how such sensor data can be used to detect and classify camera movement, shaking, or orientation. Such contextual information can be useful to deciding if a video sequence is of good quality. Using multiple cameras recording the same scene, the authors demonstrate how data of auxiliary sensors can be utilized to obtain contextual information about the scene, such as the angular region of interest of a public happening. By correlating the movement of multiple cameras, Cricri et al. show the extraction of contextual information about events, such as the lifting of hands in an audience.

Related to context awareness, and the recognition of context, is the problem of activity recognition, in particular the recognition of user activity in smart environments, such as smart homes, for ambient assisted living. A prominent scenario is that of assisting the elderly in their homes in order to prolong independent living. Assam and Seidl (2014) underscore that "[i]nferring high level activity or context from low level sensor signals has sparked huge research interest to discover emerging patterns and correlations from sensor data." According to Assam and Seidl, a challenging task in this problem space is to define the signature feature vector corresponding to an activity and thus to "detect the dissimilarity between signals of two different activities." Assam and Seidl present a model capable of predicting high level activity from low level accelerometer sensor signals stemming from smart phones. The predicted activities include jogging, walking, ascending stairs, descending stairs, sitting, and standing.

Ye et al. (2014) propose an unsupervised semantic mining activity recognition technique, which the authors present "as a systematic way of deeply integrating knowledge- and data-driven techniques." The technique includes a general ontological framework consisting of object, location, sensor, and activity ontolo-

gies. The sensor ontology is intended for the representation of sensor events modelled as tuples consisting of values for time, sensor id, and reported value. In their experiments, sensors observe state changes and values are either 0 or 1, e.g. for closed or open door. The monitored activities include, among other, leaving the house, taking a shower, using the toilet, and going to bed. Thus, the technique recognizes activity that involves user-object interaction. Based on the object to which a sensor is attached and the location at which the sensor is installed, ontologies determine the key sensors associated to activities. Ontologies also serve in the computation of the semantic similarity between sensor events, a measure used in the segmentation of continuous sensor events. The result of segmentation are sensor event sequences. Two consecutive sensor events that are considered semantically similar are part of the same sequence. Sequences are then mapped to activities based on key sensors in each sequence. For each activity, the set of identified sequences is then clustered using k-means in order to obtain the k most representative sensor event sequences for the activity. Given a continuous stream of sensor events, the technique can segment the stream and then classify the obtained sequences to activities by matching sequences with the k representatives of each event.

Chen et al. (2014) present an approach to activity modelling that "combines domain knowledge based model specification and data-driven model learning." Activity models go beyond activity recognition as they support more advanced features such as activity prediction or abnormal activity detection. Chen et al. argue that the hybrid approach enables the formulation of generic activity models, suitable for all users, using knowledge-driven methods, such as formal knowledge acquisition about daily routine activities, and the creation of individual activity models using data-driven incremental learning. The proposed process consists of three phases. In the first phase, preliminary activity models are built using prior knowledge. These preliminary models are used in the second phase to classify sensor data and recognize activities. Preliminary models are incomplete and unable to recognize all activities, due to differences in how individual users perform activities. In the third phase, the outputs of the second phase are used to learn new activities and user activity profiles. New activities are those not recognized in the second phase and reflecting frequent and semantically similar sensor data sequences. User activity profiles model the specific way a user performs activities, and include for instance the time it takes for an individual user to perform a particular activity. Chen et al. demonstrate their approach for single-user and single-activity scenarios, which are arguably simplistic compared to real situations.

Sensors are also at the base of the Internet of Things (Atzori et al., 2010, IoT), which thus faces similar challenges in sensor data processing as well as knowledge extraction from data. Atzori et al. describe the various visions and definitions that exist for the IoT paradigm, including semantic-oriented visions. The

'things' are communication enabled objects such as sensors or mobile phones capable of interaction and cooperation. These objects are expected to grow in numbers and diversity to include 'everyday things' such as food packages, furniture, or clothes. Semantic technologies could play a role in the representation, persistence, organization, and search of information generated by the IoT. Atzori et al. underscore that sensor networks will play an important role in the IoT, including in environmental monitoring applications and intelligent transportation systems. Semantic knowledge acquisition is of interest also to the IoT (Ganz et al., 2014).

5.7 STRENGTHS

The Wavellite software framework and the discussed applications have several strengths. Some are briefly described in this section. The following section will present the limitations.

A strength of Wavellite is its support for situational knowledge acquisition from the streams of numbers resulting in measurement, i.e. its support for processes that begin with raw sensor data and end with represented situational knowledge. In practice, the numbers resulting in measurement may not have any associated metadata, including timestamps, and are arguably the most basic output of (digital) sensing devices. Indeed, streams of numbers void of metadata are precisely the output of the accelerometer sensing devices used in papers **I** and **II**. The application discussed in Paper **III** builds on timestamped matrices of numbers, available as text files. Only the application discussed in Paper **IV** relies on data and rich metadata served by the INSPIRE (EU, 2007) compliant FMI Open Data service. We think that systems and architectures designed to extract knowledge from sensor data should avoid making assumptions about the input data other than that such data are numbers. Some architectures we reviewed assume that the input data are available annotated with rich metadata, encoded using advanced XML or RDF data models, or as symbolic abstractions for events of interest to the application (e.g. Loke (2004); Whitehouse et al. (2006); Janowicz (2012); Llaves and Kuhn (2014)). Some applications may receive such input data but this cannot be generalized. For applications that build on sensor network data, the conservative assumption is that data are numbers. Therefore, we argue that architectures which do not make further assumptions and support the representation of knowledge acquired from processed numbers obtained in measurement are more credible in their attempt to close the gap between high-level knowledge and low-level environmental monitoring data.

Another strength of the Wavellite software framework is that it does not prescribe a particular set of techniques or models for data processing and knowledge acquisition. Papers **II** and **III** employ methods in digital signal processing and machine learning while Paper **IV** utilizes a physically-based environmental

model. As it defers more program logic implementation to applications, this design choice may be seen as a limitation. For applications in environmental science we argue it is a strength and, to our understanding, a necessity as it can hardly be circumvented, for at least the following two reasons. First, the spectrum of possible knowledge acquisition problems on environmental sensor network data, as well as the spectrum of data processing and knowledge acquisition techniques such problems rely on, is huge. It is arguably difficult to envision an (automated) approach that fits them all. Second, program logic of software libraries, such as the environmental model used in Paper **IV**, that implement one or more, possibly complex, subtasks of the overall data processing and knowledge acquisition workflow may exist and can thus be reused. We consider it a strength if a framework supports the reuse of such program logic.

A further strength is the design choice of a hybrid approach that supports both inductive data-driven and deductive knowledge-driven techniques. As the discussed applications arguably demonstrate, both inductive and deductive techniques can assume important roles in knowledge acquisition and processing. Restricting the framework to one or the other set of techniques would inherently limit its capabilities and thus its ability to support practical problems in environmental monitoring. The hybrid approach has received support in the literature, e.g. by Janowicz et al. (2015).

Further strengths of the framework are its clear separation of the three abstraction levels, and the shared set of technologies common to data and knowledge curation and access. The discussed applications suggest that distinguishing sensor observations, dataset observations, and situations is a reasonable design choice. The three information objects are fundamental to systems designed to process and manage raw sensor data, processed data, and acquired situational knowledge. Together with associated ontologies, the three information objects also reflect the progressive level of abstraction from raw sensor data to situational knowledge. Sharing a single data model for information objects as well as the technologies common to data and knowledge curation and access are also strengths of the Wavellite framework. Having one data model (RDF), one knowledge representation language (OWL), one knowledge base (Stardog), and one query language (SPARQL) greatly simplifies the overall architecture.

A strength of the applications discussed in Chapter 4 is that they include challenging knowledge acquisition problems that rely on complex data processing, involving techniques in digital signal processing as well data-driven and physically-based modelling. Scientific applications arguably demand that software systems are capable of supporting such complexity as well as the integration of diverse techniques and models used in knowledge acquisition workflows. Knowledge acquisition by means of rules encoded in languages such as SWRL and evaluated using off-the-shelf reasoners is elegant but unfortunately address only a narrow band of relatively simplistic knowledge acquisition problems, in

the spectrum of possible problems. For practical utility, frameworks should support the implementation of problems that span as wide a spectrum as possible.

5.8 LIMITATIONS

Little thought and development has been invested so far toward user friendliness of the Wavellite framework. While setting up the development environment and required software, such as an RDF database, are relatively trivial tasks, one of the most important limitation of Wavellite in its current form is that in order to implement a particular application users are required to extend the framework with knowledge and program logic. Users are thus required to develop (Java) program logic. For instance, after having created a training dataset and evaluated the performance of a machine learning technique for a particular classification task, it is necessary to develop program logic that integrates the trained classifier in a workflow for knowledge acquisition, as well as program logic that implements and executes the workflow.

Writing such program logic is often expensive and cumbersome. Application developers are required to be fluent in programming and familiar with specialized methods in data processing, data mining, and knowledge representation and reasoning. Such a set of skills is arguably uncommon among domain experts, such as environmental scientists. Thus, the development of Wavellite applications currently requires collaboration between scientists and software engineers. Writing program logic is also error prone and it does not scale well to large numbers of knowledge acquisition problems. Furthermore, the knowledge acquisition process is implicit in program code.

As hinted in sections 5.2 and 5.3, systems can adopt an abstract concept other than that of situation, as understood in situation theory, to organize knowledge acquired from data. Situation theory and its formalization of situation and information about situations is arguably a reasonable candidate within the framework of situation awareness, for at least two reasons. First, the notion of situation is obviously central to situation awareness, in particular also the situation theoretic formalization (Kokar and Endsley, 2012). Second, in situation assessment data and information are processed from perception to projection and situation theory provides a needed formal framework for the representation of information about situations. However, a comparative evaluation of the implications of adopting one concept over the other is certainly interesting and this dissertation has put little emphasis on this aspect.

Situational knowledge is generally situated in space and time: it is assertional knowledge about individual situations. The acquisition of terminological knowledge, currently not addressed by Wavellite, is also relevant to scientific workflows. For instance, the discovery of new particle formation as a natural phenomenon results in the definition of a new situation type for the class of new

particle formation events. Stocker et al. (2011) discuss a case where the threshold values in atoms of rules for the classification of nutrient rich and nutrient poor lakes are acquired from sensor data. Rules are terminological knowledge. Work by other authors focuses on terminological knowledge acquisition (Maedche and Staab, 2001; Janowicz, 2012). Findings in environmental research can often be understood as terminological knowledge because findings are often for categories. The statements (i) lakes with high productivity are eutrophic and (ii) acute exposure of plants to ozone can induce cell death, involve categories and are not about a particular lake or plant. Software support for the curation, access, and processing of terminological knowledge acquired from processed data is therefore of interest to environmental research. For environmental research infrastructure it may be of interest to support both terminological and assertional knowledge acquisition. Our focus in this dissertation is on assertional knowledge.

In Wavellite applications, sensor observations, dataset observations, and situations are managed by the RDF database. An RDF database, or 'triple store', is arguably not designed for the management of large volumes of sensor and dataset observations. This concern was also highlighted by other authors, e.g., Le-Phuoc et al. (2011). Representing a sensor observation in RDF conforming with the SSN ontology results in approximately a dozen triples (the actual number can vary). Converting a time and value pair into a sensor observation thus results in a considerable increase of data. This increase can be attributed in part to metadata about sensor, property, and feature being associated with individual observations, and in part to the RDF data model—in particular also to the use of URI, each of which can be of considerable length. Furthermore, queries for observations typically match a predictable pattern. For instance, a frequent query for sensor observations specifies the sensor, property, feature, and a time interval. Such a query can be evaluated efficiently on large data volumes provided that the data are indexed effectively and organized along the temporal dimension. RDF databases are required to optimize the evaluation of arbitrary queries. Hence, they maintain generic indexes on RDF subjects, predicates, and objects (Weiss et al., 2008). While obviously necessary for SPARQL query evaluation, the generic design is arguably suboptimal for sensor and dataset observation query evaluation.

In building on ENVRI-RM, we have aligned this dissertation with environmental research infrastructure and argued that some of the design choices, such as the explicit representation of sensor observations and datasets, are relevant to environmental research infrastructure more than to related areas, such as robotics or ambient intelligence, in which knowledge acquisition from processed sensor data is also studied. However, only Paper **III** on situational knowledge for atmospheric phenomena is a clear case study in environmental research infrastructure. The case study of Paper **IV** on plant disease pressure situation

modelling serves primarily two communities: farmers and agricultural advisers. The latter community consists of researchers in agricultural science. Farmers, on the other hand, are primarily consumers of situational knowledge. Hence, the case study of Paper **IV** serves a more heterogeneous community. In contrast, the case study of Paper **II** on road traffic is closer to civil engineering. The argued alignment of this dissertation with environmental research infrastructure would arguably benefit from further case studies with focus on environmental research infrastructure.

The alignment with environmental research infrastructure occurred late in the development of this dissertation. It is a consequence of our attempt to ground the Wavellite architecture into an existing and suitable reference model, an effort we deem valuable because it facilitates understanding and communication. Furthermore, the alignment is interesting because, compared to modelling the data life-cycle, modelling the curation, access, and processing of (situational) knowledge acquired from processed (sensor) data is relatively novel in environmental research infrastructure. With the alignment we underscore that environmental research infrastructures may go beyond data life-cycle management to also support information and knowledge life-cycle management.

However, the systems of primary interest to the research question are environmental monitoring systems. Environmental monitoring is fundamental to environmental research as well as to applications in other domains, such as civil engineering. The discussed case studies are thus within the scope of the research question, and provide evidence that situation theory and methods in ontology engineering can be utilized by environmental monitoring systems in heterogeneous applications, possibly applications of environmental research infrastructure.

The focus of this work on *in situ* environmental sensor networks, and sensor observations with observation value corresponding to a number, is a further limitation. Remote sensing is an important alternative to *in situ* sensor networks in environmental monitoring. Furthermore, the observation value may be data other than a number. For instance, a sensing device attached to a satellite may perform remote sensing, and observation values are multidimensional arrays of numbers.

5.9 FUTURE WORK

There exist several directions for future work, which may be guided by the limitations discussed in the previous section. In this section, we propose and discuss in further details five directions, namely (1) the development of a database management system for the persistence of RDF sensor and dataset observations; (2) the possibility of utilizing (semantic) workflow systems to describe data processing and knowledge acquisition and representation in Wavellite; (3) the ex-

tension of the knowledge layer with types other than situation; (4) advancing knowledge-based environmental research infrastructure; and (5) developing the approach in industry trends such as smart homes, smart grids, and smart cities.

RDF databases are ill-suited for the management of RDF data that are ordered in time, such as sensor observations. This is because a triple store is fundamentally an unordered set of RDF statements, i.e. triples. The evaluation of a typical SSN observation query that constrains the sensor, property, feature, and time interval, is an expensive operation for RDF databases, and quickly prohibitive for time series with a few million sensor observations. This is the case also for dataset observations, though it is less expensive to evaluate a query for multidimensional QB observations than to evaluate a comparable query for SSN observations with properties and features corresponding to the dimensions of the queried QB observations. This is because the values of several dimensions are directly related to QB observations whereas the values of different properties and features are related to *different* SSN observations.

The inability of RDF databases to scale to large time series arguably constitutes a practical engineering problem in management of sensor and dataset observations encoded in RDF, possibly according to the SSN ontology and the QB vocabulary, respectively. We are currently developing a database management system, called Emrooz (Stocker et al., 2015c), designed to consume and return SSN observations, and expected to support fast evaluation of SSN observation queries on time series with several billion sensor observations. Database management systems that support the curation of and access to very large time series represented in RDF, specifically following the SSN ontology, are of crucial importance to applications, in particular also Wavellite applications. Emrooz builds on Apache Cassandra to persist temporally ordered SSN observations and evaluates SSN observation queries formulated in SPARQL. NoSQL databases, such as Cassandra, have been advocated and developed for the management of RDF data (Cudré-Mauroux et al., 2013), including Cassandra (Ladwig and Harth, 2011). Our aim is different from such efforts because we tailor Emrooz for SSN observations, rather than for generic RDF data.

We also envision Emrooz support for datasets and dataset observations encoded in RDF following the QB vocabulary. Emrooz could thus support the efficient persistence and retrieval of large volumes of sensor and dataset observations in (Wavellite) applications. In addition to data ordered in time, it will be interesting to also explore the possibility of managing data ordered in space using Cassandra. Finally, the current Emrooz API could be expended to support functionality beyond adding and querying observations to include functionality for processing observations. Examples include simple operations such as computing hourly average of dataset observations.

The second direction for future work is to study the possibility of adopting a semantic workflow system for the declarative formulation of data process-

ing and knowledge acquisition and representation, as well as the processing of knowledge, in Wavellite applications. Considerable research effort has been devoted toward semantic workflow systems. The resulting literature is interesting to Wavellite workflow modelling, and the modelling of workflows for data processing and knowledge acquisition and representation in situation-aware environmental monitoring systems more generally. To name a few examples, Deelman et al. (2005) and Gil et al. (2011) present the Wings/Pegasus intelligent workflow system, which aims at assisting users with designing computational experiments as workflow. The system supports both workflow creation and execution and addresses concerns such as workflow validation and workflow mapping to computing resources. Ludäscher et al. (2006) present the Kepler system for web service-based scientific workflow management. Wings/Pegasus and Kepler are potentially interesting for the user-friendly formulation of data flow and processing in Wavellite layers, particularly at the derivation layer where chained derivation engines transform the data of input datasets into output datasets. As in these scientific workflow systems, Wavellite components can be modelled as the computational nodes of a directed acyclic graph, with directed edges representing data flow.

The third direction is to extend the knowledge layer of the Wavellite architecture with conceptual types other than situation. Of particular interest is the concept of process, and Galton and Mizoguchi (2009) provide a useful starting point with a discussion on ontological modelling of processes. The extension of the knowledge layer with further abstract types could improve the applicability of the proposed framework to new problems and domains. For instance, in greenhouse gas monitoring, devices measure surface-atmosphere fluxes of energy and trace gases. It is arguably more appropriate to understand such flux as a process, open-ended and able to undergo change, rather than a situation or an event. Knowledge obtained from data about CO_2 fluxes between the canopy of a forest and the atmosphere is thus knowledge about a process. Extending the knowledge layer with support for this kind could enable the representation of such knowledge and the application of the framework to new problems.

The fourth direction is to evolve state of the art environmental research infrastructure into knowledge-based systems. Based on ENVRI-RM and related research, we have highlighted that state of the art environmental research infrastructure is primarily concerned with data life-cycle management. Though the data processing subsystem includes functionality for data analysis and mining, ENVRI-RM does not specify what occurs to information and knowledge obtained in such data processing. In other words, ENVRI-RM does not model the knowledge life-cycle in environmental research infrastructure and is not concerned with information and knowledge life-cycle management. In Paper **V**, we have made the case for knowledge-based environmental research infrastructure, i.e. infrastructure that manages the life-cycles of data required in interpretation

and information and knowledge gained from data. This dissertation highlights situational knowledge as one possible knowledge type of interest to environmental research infrastructure. It also details technologies, and their integration in systems, that may be of interest to knowledge-based environmental research infrastructure.

However, significantly more work is needed to evolve state of the art environmental research infrastructure into knowledge-based systems. Such an endeavour relies on strong collaboration between environmental research infrastructure projects and research groups with expertise in knowledge-based systems, artificial intelligence, and semantic web technologies. We have recently initiated a collaboration with ICOS Carbon Portal and plan to collaborate with other projects in the future. If successful, next generation environmental research infrastructure could be more than mere web portals for downloading or visualizing data, as is typical today. In addition to observational data, ICOS Carbon Portal aims at managing also elaborated data products, such as map visualizations for CO_2 fluxes, which the science community can submit to the platform. These are interesting steps toward environmental research infrastructure with functionality beyond data download. However, information remains implicit in maps and is thus not accessible and unavailable for automated management and processing by the infrastructure. Looking at a European map for greenhouse gas fluxes, a trained scientist may instantly recognize major European cities as strong greenhouse gas sources. As this information is implicit in the colour scale and map features, this inference is non-trivial for the software components of environmental research infrastructure. Ideally, these components would also have access to such information and would be able to interact with the science community of environmental research infrastructure using high level concepts such as city or strong source.

This dissertation, in particular Paper **III** and Paper **IV**, has demonstrated how information about monitored phenomena extracted from data managed by environmental research infrastructure can be modelled as situational knowledge, and how such knowledge can be represented explicitly in infrastructure. Explicit representation enables management, processing, visualization, reasoning, integration, and sharing of situational knowledge.

The collaboration with existing environmental research infrastructures will also provide interesting new case studies. They will highlight new requirements and demonstrate the modelling of situational knowledge with more complex structure and information extracted from multiple knowledge acquisition processes on heterogeneous data.

Finally, as a fifth direction for future work, we propose that the ideas discussed in this dissertation can also be applied to industry trends such as smart homes, smart grids, and smart cities. Systems in these domains often build on sensor networks to monitor environments and infrastructure. Such monitoring

results in (streamed) data, typically of considerable volume and heterogeneity. Common with the applications discussed in Chapter 4 is the need to obtain from sensor data symbolic knowledge about the monitored environment and infrastructure, represented using the high level concepts and relations familiar to agents, in particular human agents. A smart home could thus schedule a laundry turn according to water and electricity price considerations of home inhabitants; a smart grid could help building and maintaining the situation awareness of technicians for the state of the grid; a smart city with commuting modes ranging from bicycle to car pooling may inform inhabitants about, and automatically schedule, the ideal means of transportation depending on current and projected situation awareness, formed by information about the weather, neighbours travelling on a similar route, route safety, etc.

6 Conclusion

We set forth with the aim of demonstrating how environmental monitoring systems can utilize situation theory to model observed situations, and utilize ontology and related technologies in knowledge representation and reasoning to represent situational knowledge obtained from data processed by means of computational models.

Building on existing concepts and technologies in environmental monitoring, situation awareness, situation theory, ontology, and modelling we have proposed an architecture and implementation for a software framework designed to support the development of environmental monitoring systems that process environmental sensor network data, acquire situational knowledge from data, and curate and process situational knowledge. The software framework served in the development of applications in case studies for environmental monitoring in intelligent transportation systems, atmospheric science, and agricultural science.

Each application is for an environmental monitoring system consisting of a monitored environment, hardware and software components, and human experts. They are thus physical-socio-technical systems with the monitored environment as the physical subsystem, human experts as the social subsystem, and hardware and software components as the technical subsystem. Each environmental monitoring system observes situations of the monitored environment and uses situation theory to model observed situations. The applications thus validate Claim **C1**.

The environmental monitoring systems process data acquired from environmental sensor networks using computational models, and utilize the STO, OWL-Time, and GeoSPARQL ontologies as well as ontology languages, software libraries for knowledge representation and reasoning, and knowledge base technologies to represent situational knowledge obtained from processed data. The applications thus validate Claim **C2**.

The development of the environmental monitoring system in each application is supported by a software framework for situation awareness in environmental monitoring. The layered framework architecture supports fundamental functionality for the perception, comprehension, and projection of situations and involved phenomena. Perception builds on environmental sensor networks, and is implemented at the measurement layer. Comprehension builds on computational data processing at the observation and derivation layers, and is implemented at the situation layer using computational knowledge acquisition. Situation projection in the near future builds on processed data and represented situational knowledge, and is implemented at the situation or processing layers using knowledge acquisition or reasoning. The environmental monitoring sys-

tems thus obtain and maintain situation awareness via the process of situation assessment. We speak of situation-aware environmental monitoring systems. The applications thus support Claim **C3**.

The software framework architecture is grounded in the ENVRI reference model for 'archetypical' environmental research infrastructure extended with a model for the acquisition of knowledge from data, and the curation, access, and processing of knowledge. The case studies in atmospheric science and in agricultural science include environmental monitoring systems with social subsystems consisting of science communities. The systems are examples of knowledge-based environmental research infrastructure designed to acquire data from environmental sensor networks, curate and process data, acquire situational knowledge from processed data, and curate and process situational knowledge. The applications thus support Claim **C4**.

The main contributions are the ENVRI reference model extension; the architecture of a software framework for situation awareness in environmental monitoring; the open source implementation for the software framework architecture; and the evaluation and discussion of the software framework implementation for three case studies with environmental monitoring systems for situational knowledge acquisition and processing in intelligent transportation systems, atmospheric science, and agricultural science.

An important strength of the proposed software framework is its ability to support the development of systems that are required to implement complete data and knowledge life-cycles, starting with the digital numbers collected from sensing devices and ending with processed situational knowledge curated by knowledge bases. A second important strength is the adoption of a hybrid approach that supports both inductive data-driven and deductive knowledge-driven techniques. A third strength of this work is the demonstration of the proposed approach in case studies with non-trivial data processing and knowledge acquisition problems. The main limitations of the software framework are the considerable resources and expertise required to develop applications, the omission of terminological knowledge acquisition, and the disregard of techniques in environmental monitoring other than *in situ* environmental sensor networks.

The future is bright for further work along the lines drawn by this dissertation. Much research and development can be pursued to evolve state of the art data-based environmental research infrastructure toward knowledge-based systems. This multidisciplinary endeavour is interesting and challenging, and rests on strong collaboration between disciplines and state of the art environmental research infrastructure projects. Such endeavour will highlight the technical components that require further development in order to meet expectations, such as the efficient persistence and retrieval of large volumes of sensor and dataset observations encoded in RDF. The effort will also highlight the potential of the approach in large scale environmental research infrastructure.

Tollefson (2011) underscores that NEON has battled scepticism. He refers to Steven Wofsy, "a pioneer of carbon studies [who] remains sceptical of big science projects," and recollects that Wofsy "feared that NEON would generate more data than value." The systems for data acquisition, curation, access, and processing aimed at in projects such as ICOS and NEON build an invaluable foundation for big science research. After all, the data are arguably a precondition to addressing the "really big ecological questions" referred to by Wofsy. However, the systems and models such as the ENVRI reference model need to push the boundary beyond data, beyond web portals from which science communities can download data. The systems should actively support science communities in knowledge acquisition, attempt to automate such tasks, and handle knowledge curation and access. Systems can then automatically process curated knowledge, a capability that may enable systems to also support science communities in hypothesis formulation and testing. To get there, the disciplines need to team up. Unless computer scientists and software engineers with expertise in computational intelligence and knowledge engineering, and artificial intelligence more broadly, join the efforts embarked by projects such as ICOS and NEON to develop next generation knowledge-based environmental research infrastructure, the systems will likely remain at the stage of web portals from which science communities can download data. In such event, Wofsy may then correctly claim "I told you so."

Bibliography

Aamodt, A. and Nygård, M. (1995). Different roles and mutual dependencies of data, information, and knowledge – An AI perspective on their integration. *Data & Knowledge Engineering*, 16(3):191–222.

Abadi, D. J., Ahmad, Y., Balazinska, M., Çetintemel, U., Cherniack, M., Hwang, J.-H., Lindner, W., Maskey, A. S., Rasin, A., Ryvkina, E., Tatbul, N., Xing, Y., and Zdonik, S. (2005). The Design of the Borealis Stream Processing Engine. In *Second Biennial Conference on Innovative Data Systems Research (CIDR 2005)*, pages 277–289, Asilomar, CA.

Abecker, A., Brauer, T., Magoutas, B., Mentzas, G., Papageorgiou, N., and Quenzer, M. (2014). A Sensor and Semantic Data Warehouse for Integrated Water Resource Management. In Gómez, J. M., Sonnenschein, M., Vogel, U., Winter, A., Rapp, B., and Giesen, N., editors, *Proceedings of the 28th Conference on Environmental Informatics - Informatics for Environmental Protection, Sustainable Development and Risk Management (EnviroInfo 2014)*, pages 517–524, Oldenburg, Germany. BIS-Verlag, Oldenburg.

Ackland, R., Gouweleeuw, B., Ticehurst, C., Thew, P., Raupach, T., and Squire, G. (2012). Blending satellite observations to provide automated monitoring of flood events. In Grove, J. and Rutherford, I., editors, *Proceedings of the 6th Australian Stream Management Conference, Managing for Extremes*, pages 197–205, Canberra, Australia. River Basin Management Society.

Ahmedi, L., Jajaga, E., and Ahmedi, F. (2013). An Ontology Framework for Water Quality Management. In Corcho, O., Henson, C., and Barnaghi, P., editors, *Proceedings of the 6th International Workshop on Semantic Sensor Networks*, volume 1063, pages 35–50, Sydney, Australia. CEUR-WS.

Akyildiz, I., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002). A survey on sensor networks. *Communications Magazine, IEEE*, 40(8):102–114.

Alirezaie, M. and Loutfi, A. (2014). Reasoning for Improved Sensor Data Interpretation in a Smart Home. In *Proceedings of the 6th International Workshop on Acquisition, Representation and Reasoning about Context with Logic (ARCOE-Logic 2014)*, Linköping, Sweden.

Assam, R. and Seidl, T. (2014). Activity Recognition from Sensors using Dyadic Wavelets and Hidden Markov Model. In *Wireless and Mobile Computing, Networking and Communications (WiMob), 2014 IEEE 10th International Conference on*, pages 442–448.

Attard, J., Scerri, S., Rivera, I., and Handschuh, S. (2013). Ontology-based Situation Recognition for Context-aware Systems. In *Proceedings of the 9th International Conference on Semantic Systems*, I-SEMANTICS '13, pages 113–120, New York, NY, USA. ACM.

Atzori, L., Iera, A., and Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks*, 54(15):2787–2805.

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. (2007). DBpedia: A Nucleus for a Web of Open Data. In Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., and Cudré-Mauroux, P., editors, *The Semantic Web*, volume 4825 of *Lecture Notes in Computer Science*, pages 722–735. Springer Berlin Heidelberg.

Baader, F., Calvanese, D., McGuinness, D., Nardi, D., and Patel-Schneider, P. (2007). *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2nd Edition.

Babu, S. and Widom, J. (2001). Continuous Queries over Data Streams. *SIGMOD Rec.*, 30(3):109–120.

Baca, A., Dabnichki, P., Heller, M., and Kornfeind, P. (2009). Ubiquitous computing in sports: A review and analysis. *Journal of Sports Sciences*, 27(12):1335–1346. PMID: 19764000.

Balazinska, M., Deshpande, A., Franklin, M., Gibbons, P., Gray, J., Nath, S., Hansen, M., Liebhold, M., Szalay, A., and Tao, V. (2007). Data Management in the Worldwide Sensor Web. *Pervasive Computing, IEEE*, 6(2):30–40.

Baral, C. (2003). *Knowledge Representation, Reasoning and Declarative Problem Solving*. Cambridge University Press.

Barbieri, D., Braga, D., Ceri, S., Valle, E. D., and Grossniklaus, M. (2010). Stream Reasoning: Where We Got So Far. In *Proceedings of the 4th International Workshop on New Forms of Reasoning for the Semantic Web: Scalable & Dynamic*, pages 1–7, Heraklion, Crete, Greece.

Barbieri, D. F., Braga, D., Ceri, S., Della Valle, E., and Grossniklaus, M. (2009). C-SPARQL: SPARQL for Continuous Querying. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 1061–1062, New York, NY, USA. ACM.

Barnaghi, P., Ganz, F., Henson, C., and Sheth, A. (2012). Computing Perception from Sensor Data. In *Sensors, 2012 IEEE*, pages 1–4.

Barnaghi, P., Meissner, S., Presser, M., and Moessner, K. (2009). Sense and Sens'ability: Semantic Data Modelling for Sensor Networks. In Cunningham, P. and Cunningham, M., editors, *Proceedings of the ICT Mobile Summit*. IIMC International Information Management Corporation.

Barnsley, M. J. (2007). *Environmental Modelling: A Practical Introduction*. CRC Press.

Barwise, J. and Perry, J. (1980). The Situation Underground. In Barwise, J. and Sag, I., editors, *Stanford Working Papers in Semantics*, volume 1, pages 1–55. Stanford Cognitive Science Group.

Barwise, J. and Perry, J. (1981). Situations and Attitudes. *The Journal of Philosophy*, 78(11):668–691.

Barwise, J. and Perry, J. (1983). *Situations and attitudes*. Bradford books. MIT Press.

Battams, K. (2014). Stream Processing for Solar Physics: Applications and Implications for Big Solar Data. In Angryk, R. A. and Martens, P. C., editors, *Proceedings of the 1st Workshop on Management, Search and Mining of Massive Repositories of Solar Astronomy Data*, Washington DC, USA.

Baumgartner, N., Gottesheim, W., Mitsch, S., Retschitzegger, W., and Schwinger, W. (2010). BeAware!–situation awareness, the ontology-driven way. *Data & Knowledge Engineering*, 69(11):1181–1193.

Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D. L., Patel-Schneider, P. F., and Stein, L. A. (2004). OWL Web Ontology Language Reference. Recommendation, W3C.

Beckett, D. (2004). RDF/XML Syntax Specification (Revised). Recommendation, W3C.

Benson, B. J., Bond, B. J., Hamilton, M. P., Monson, R. K., and Han, R. (2010). Perspectives on next-generation technology for environmental sensor networks. *Frontiers in Ecology and the Environment*, 8(4):193–200.

Berners-Lee, T., Cailliau, R., Luotonen, A., Nielsen, H. F., and Secret, A. (1994). The World-Wide Web. *Commun. ACM*, 37(8):76–82.

Berners-Lee, T., Fielding, R., and Masinter, L. (2005). Uniform Resource Identifier (URI): Generic Syntax. Request for Comments 3986, IETF.

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The Semantic Web. *Scientific American*, 284(5):29–37.

Beven, K. (2002). Towards an alternative blueprint for a physically based digitally simulated hydrologic response modelling system. *Hydrological Processes*, 16(2):189–206.

Bizer, C., Heath, T., and Berners-Lee, T. (2011). Linked Data: The Story so Far. In Sheth, A., editor, *Semantic Services, Interoperability and Web Applications: Emerging Concepts*, pages 205–227. IGI Global, Hershey, PA.

Bolles, A., Grawunder, M., and Jacobi, J. (2008). Streaming SPARQL - Extending SPARQL to Process Data Streams. In Bechhofer, S., Hauswirth, M., Hoffmann, J., and Koubarakis, M., editors, *The Semantic Web: Research and Applications*, volume 5021 of *Lecture Notes in Computer Science*, pages 448–462. Springer Berlin Heidelberg.

Bonino, D. and Corno, F. (2012). spChains: A Declarative Framework for Data Stream Processing in Pervasive Applications. *Procedia Computer Science*, 10:316–323.

Borst, W. N. (1997). *Construction of Engineering Ontologies for Knowledge Sharing and Reuse*. PhD thesis, Centre for Telematics and Information Technology, University of Twente, Enschede, The Netherlands.

Botts, M., Percivall, G., Reed, C., and Davidson, J. (2007). OGC© Sensor Web Enablement: Overview And High Level Architecture. OGC White Paper OGC 07-165, Open Geospatial Consortium Inc.

Botts, M., Percivall, G., Reed, C., and Davidson, J. (2008). OGC© Sensor Web Enablement: Overview and High Level Architecture. In Nittel, S., Labrinidis, A., and Stefanidis, A., editors, *GeoSensor Networks*, volume 4540 of *Lecture Notes in Computer Science*, pages 175–190. Springer Berlin Heidelberg.

Botts, M. and Robin, A. (2007). OpenGIS Sensor Model Language (SensorML) Implementation Specification. OpenGIS Implementation Specification OGC 07-000, Open Geospatial Consortium Inc.

Bray, T., Paoli, J., and Sperberg-McQueen, C. M. (1998). Extensible Markup Language (XML) 1.0. Recommendation, W3C.

Brickley, D. and Guha, R. (2004). RDF Vocabulary Description Language 1.0: RDF Schema. Recommendation, W3C.

Brickley, D. and Guha, R. (2014). RDF Schema 1.1. Recommendation, W3C.

Bröring, A., Maué, P., Malewski, C., and Janowicz, K. (2012). Semantic mediation on the Sensor Web. In *Geoscience and Remote Sensing Symposium (IGARSS), 2012 IEEE International*, pages 2910–2913.

Calbimonte, J.-P., Corcho, O., and Gray, A. J. (2010). Enabling Ontology-Based Access to Streaming Data Sources. In Patel-Schneider, P. F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J. Z., Horrocks, I., and Glimm, B., editors, *The Semantic Web – ISWC 2010*, volume 6496 of *Lecture Notes in Computer Science*, pages 96–111. Springer Berlin Heidelberg.

Calbimonte, J.-P., Yan, Z., Jeung, H., Corcho, O., and Aberer, K. (2012). Deriving Semantic Sensor Metadata from Raw Measurements. In Henson, C., Taylor, K., and Corcho, O., editors, *Proceedings of the 5th International Workshop on Semantic Sensor Networks*, volume 904, pages 33–48, Boston, Massachusetts, USA. CEUR-WS.

Cappellari, P., Shi, J., Roantree, M., Tobin, C., and Moyna, N. (2011). Enabling Knowledge Extraction from Low Level Sensor Data. In Hameurlain, A., Liddle, S., Schewe, K.-D., and Zhou, X., editors, *Database and Expert Systems Applications*, volume 6861 of *Lecture Notes in Computer Science*, pages 411–419. Springer Berlin Heidelberg.

Carney, D., Çetintemel, U., Cherniack, M., Convey, C., Lee, S., Seidman, G., Stonebraker, M., Tatbul, N., and Zdonik, S. (2002). Monitoring streams: a new class of data management applications. In *Proceedings of the 28th international conference on Very Large Data Bases*, VLDB '02, pages 215–226. VLDB Endowment.

Carothers, G. and Seaborne, A. (2014). RDF 1.1 N-Triples: A line-based syntax for an RDF graph. Recommendation, W3C.

Castelli, G., Mamei, M., Rosi, A., and Zambonelli, F. (2009). Extracting High-Level Information from Location Data: The W4 Diary Example. *Mobile Networks and Applications*, 14(1):107–119.

Chamberlin, D. D. and Boyce, R. F. (1974). SEQUEL: A Structured English Query Language. In *Proceedings of the 1974 ACM SIGFIDET (Now SIGMOD) Workshop on Data Description, Access and Control*, SIGFIDET '74, pages 249–264, New York, NY, USA. ACM.

Chandrasekaran, S., Cooper, O., Deshpande, A., Franklin, M. J., Hellerstein, J. M., Hong, W., Krishnamurthy, S., Madden, S., Raman, V., Reiss, F., and Shah, M. A. (2003). TelegraphCQ: Continuous Dataflow Processing for an Uncertain World. In *Proceedings of the 1st Biennial Conference on Innovative Data Systems Research*, Asilomar, CA, USA.

Chang, K., Yau, N., Hansen, M., and Estrin, D. (2006). SensorBase.org – A Centralized Repository to Slog Sensor Network Data. In *Proceedings of the IEEE International Conference on Distributed Computing in Sensor Systems (IEEE DCOSS)*, San Francisco, USA.

Chaves, F., Mossgraber, J., Schenk, M., and Bübrogel, U. (2013). Semantic Registries for Heterogeneous Sensor Networks: Bridging the Semantic Gap for Collaborative Crises Management. In *Database and Expert Systems Applications (DEXA), 2013 24th International Workshop on*, pages 118–122.

Chella, A., Frixione, M., and Gaglio, S. (1997). A cognitive architecture for artificial vision. *Artificial Intelligence*, 89(1-2):73–111.

Chen, L., Nugent, C., and Okeyo, G. (2014). An Ontology-Based Hybrid Approach to Activity Modeling for Smart Homes. *Human-Machine Systems, IEEE Transactions on*, 44(1):92–105.

Chen, Y., Hardisty, A., Preece, A., Martin, P., Atkinson, M., Zhao, Z., Magagna, B., Schentz, H., and Legré, Y. (2013a). Analysis of Common Requirements for Environmental Science Research Infrastructures. In *Proceedings of the International Symposium on Grids and Clouds (ISGC)*, Academia Sinica, Taipei, Taiwan. Proceedings of Science (SISSA).

Chen, Y., Martin, P., Magagna, B., Schentz, H., Zhao, Z., Hardisty, A., Preece, A., Atkinson, M., Huber, R., and Legré, Y. (2013b). A Common Reference Model for Environmental Science Research Infrastructures. In Page, B., Fleischer, A. G., Göbel, J., and Wohlgemuth, V., editors, *27th International Conference on Environmental Informatics for Environmental Protection, Sustainable Development and Risk Management*, pages 665–673, Hamburg, Germany.

Chong, C.-Y. and Kumar, S. (2003). Sensor Networks: Evolution, Opportunities, and Challenges. *Proceedings of the IEEE*, 91(8):1247–1256.

Collins, S. L., Bettencourt, L. M., Hagberg, A., Brown, R. F., Moore, D. I., Bonito, G., Delin, K. A., Jackson, S. P., Johnson, D. W., Burleigh, S. C., Woodrow, R. R., and McAuley, J. M. (2006). New opportunities in ecological sensing using wireless sensor networks. *Frontiers in Ecology and the Environment*, 4(8):402–407.

Compieta, P., Martino, S. D., Bertolotto, M., Ferrucci, F., and Kechadi, T. (2007). Exploratory spatio-temporal data mining and visualization. *Journal of Visual Languages & Computing*, 18(3):255–279. Visual Languages and Techniques for Human-GIS Interaction.

Compton, M., Barnaghi, P., Bermudez, L., García-Castro, R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., Herzog, A., Huang, V., Janowicz, K., Kelsey, W. D., Phuoc, D. L., Lefort, L., Leggieri, M., Neuhaus, H., Nikolov, A., Page, K., Passant, A., Sheth, A., and Taylor, K. (2012). The SSN ontology of the W3C semantic sensor network incubator group. *Web Semantics: Science, Services and Agents on the World Wide Web*, 17(0):25–32.

Compton, M., Henson, C., Lefort, L., Neuhaus, H., and Sheth, A. (2009). A Survey of the Semantic Specification of Sensors. In Taylor, K., Ayyagari, A., and Roure, D. D., editors, *Proceedings of the 2nd International Workshop on Semantic Sensor Networks*, volume 522, pages 17–32, Washington DC, USA. CEUR-WS.

Conroy, K., May, G., Roantree, M., Warrington, G., Cullen, S. J., and McGoldrick, A. (2011a). Knowledge acquisition from sensor data in an equine environment. In *Proceedings of the 13th international conference on data warehousing and knowledge discovery*, DaWaK'11, pages 432–444, Berlin, Heidelberg. Springer-Verlag.

Conroy, K., May, G. C., Roantree, M., and Warrington, G. (2011b). Expanding sensor networks to automate knowledge acquisition. In *Proceedings of the 28th British national conference on Advances in databases*, BNCOD'11, pages 97–107, Berlin, Heidelberg. Springer-Verlag.

Cook, D. J. (2007). Making Sense of Sensor Data. *IEEE Pervasive Computing*, 6(2):105–108.

Coradeschi, S. and Saffiotti, A. (2000). Anchoring Symbols to Sensor Data: preliminary report. In *Proceedings of the 17th AAAI Conference*, pages 129–135.

Coradeschi, S. and Saffiotti, A. (2003). An introduction to the anchoring problem. *Robotics and Autonomous Systems*, 43(2-3):85–96.

Cox, S. (2011). Observations and Measurements - XML Implementation. OGC Implementation OGC 10-025r1, Open Geospatial Consortium Inc.

Cregan, A. M. (2007). Symbol Grounding for the Semantic Web. In Franconi, E., Kifer, M., and May, W., editors, *The Semantic Web: Research and Applications*, volume 4519 of *Lecture Notes in Computer Science*, pages 429–442. Springer Berlin Heidelberg.

Cricri, F., Dabov, K., Curcio, I. D. D., Mate, S., and Gabbouj, M. (2014). Multimodal extraction of events and of information about the recording activity in user generated videos. *Multimedia Tools and Applications*, 70(1):119–158.

Cudré-Mauroux, P., Enchev, I., Fundatureanu, S., Groth, P., Haque, A., Harth, A., Keppmann, F. L., Miranker, D., Sequeda, J. F., and Wylot, M. (2013). NoSQL Databases for RDF: An Empirical Evaluation. In Alani, H., Kagal, L., Fokoue, A., Groth, P., Biemann, C., Parreira, J., Aroyo, L.,

Noy, N., Welty, C., and Janowicz, K., editors, *The Semantic Web – ISWC 2013*, volume 8219 of *Lecture Notes in Computer Science*, pages 310–325. Springer Berlin Heidelberg.

Cyganiak, R., Reynolds, D., and Tennison, J. (2014a). The RDF Data Cube Vocabulary. Recommendation, W3C.

Cyganiak, R., Wood, D., and Lanthaler, M. (2014b). RDF 1.1 Concepts and Abstract Syntax. Recommendation, W3C.

Dao, M.-S., Pongpaichet, S., Jalali, L., Kim, K., Jain, R., and Zettsu, K. (2014). A Real-time Complex Event Discovery Platform for Cyber-Physical-Social Systems. In *Proceedings of International Conference on Multimedia Retrieval*, ICMR '14, pages 201:201–201:208, New York, NY, USA. ACM.

De Paola, A., Gaglio, S., Lo Re, G., and Ortolani, M. (2009). An Ambient Intelligence Architecture for Extracting Knowledge from Distributed Sensors. In *Proceedings of the 2Nd International Conference on Interaction Sciences: Information Technology, Culture and Human*, ICIS '09, pages 104–109, New York, NY, USA. ACM.

Decker, S., Melnik, S., van Harmelen, F., Fensel, D., Klein, M., Broekstra, J., Erdmann, M., and Horrocks, I. (2000). The Semantic Web: The Roles of XML and RDF. *Internet Computing, IEEE*, 4(5):63–73.

Deelman, E., Singh, G., Su, M.-H., Blythe, J., Gil, Y., Kesselman, C., Mehta, G., Vahi, K., Berriman, G. B., Good, J., Laity, A., Jacob, J. C., and Katz, D. S. (2005). Pegasus: A framework for mapping complex scientific workflows onto distributed systems. *Scientific Programming*, 13(3):219–237.

Della Valle, E., Ceri, S., van Harmelen, F., and Fensel, D. (2009). It's a Streaming World! Reasoning upon Rapidly Changing Information. *Intelligent Systems, IEEE*, 24(6):83–89.

D'Este, C., Morshed, A., and Dutta, R. (2014). Robot Sensor Data Interoperability and Tasking with Semantic Technologies. In *Instrumentation and Measurement Technology Conference (I2MTC) Proceedings, 2014 IEEE International*, pages 1543–1547.

Devaraju, A. and Kauppinen, T. (2012). Sensors Tell More than They Sense: Modeling and Reasoning about Sensor Observations for Understanding Weather Events. *International Journal of Sensors Wireless Communications and Control*, 2(1):14–26.

Devaraju, A., Kuhn, W., and Renschler, C. S. (2014). A formal model to infer geographic events from sensor observations. *International Journal of Geographical Information Science*, pages 1–27.

Devlin, K. (1991). *Logic and Information*. Cambridge University Press.

Devlin, K. (2004). Jon Barwise's Papers on Natural Language Semantics. *The Bulletin of Symbolic Logic*, 10(1):54–85.

Devlin, K. (2006). Situation theory and situation semantics. In Gabbay, D. M. and Woods, J., editors, *Logic and the Modalities in the Twentieth Century*, volume 7 of *Handbook of the History of Logic*, pages 601–664. North-Holland.

Dey, A. K. (2001). Understanding and Using Context. *Personal and Ubiquitous Computing*, 5(1):4–7.

Doulaverakis, C., Konstantinou, N., Knape, T., Kompatsiaris, I., and Soldatos, J. (2011). An Approach to Intelligent Information Fusion in Sensor Saturated Urban Environments. In *Intelligence and Security Informatics Conference (EISIC), 2011 European*, pages 108–115.

Dow, A. K., Dow, E. M., Fitzsimmons, T. D., and Materise, M. M. (2014). Harnessing the Environmental Data Flood: A Comparative Analysis of Hydrologic, Oceanographic, and Meteorological Informatics Platforms. *Bulletin of the American Meteorological Society*. (In Press).

Dürst, M. and Suignard, M. (2005). Internationalized Resource Identifiers (IRIs). RFC 3987, IETF.

Egenhofer, M. J. (2002). Toward the Semantic Geospatial Web. In *Proceedings of the 10th ACM International Symposium on Advances in Geographic Information Systems*, GIS '02, pages 1–4, New York, NY, USA. ACM.

Endsley, M. R. (1995). Toward a theory of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(1):32–64.

Endsley, M. R. (1996). Automation and Situation Awareness. In Parasuraman, R. and Mouloua, M., editors, *Automation and human performance: Theory and applications*, pages 163–181. Lawrence Erlbaum Associates, Inc., Mahwah, NJ.

Endsley, M. R. (2000). Theoretical underpinnings of situation awareness: A critical review. In Endsley, M. and Garland, D. J., editors, *Situation awareness analysis and measurement*, pages 3–27. Lawrence Erlbaum Associates, Inc., Mahwah, NJ.

ENVRI (2013). ENVRI Reference Model V1.1. Technical Report RDTI-RI-283465, ENVRI.

Esling, P. and Agon, C. (2012). Time-series Data Mining. *ACM Comput. Surv.*, 45(1):12:1–12:34.

EU (2007). Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 establishing an Infrastructure for Spatial Information in the European Community (INSPIRE).

Fayyad, U. M., Piatetsky-Shapiro, G., and Smyth, P. (1996). From Data Mining to Knowledge Discovery: An Overview. In Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R., editors, *Advances in Knowledge Discovery and Data Mining*, pages 1–34. American Association for Artificial Intelligence, Menlo Park, CA, USA.

Fernández, J., Calavia, L., Baladrón, C., Aguiar, J. M., Carro, B., Sánchez-Esguevillas, A., Alonso-López, J. A., and Smilansky, Z. (2013). An Intelligent Surveillance Platform for Large Metropolitan Areas with Dense Sensor Deployment. *Sensors*, 13(6):7414–7442.

Fernández, J. D., Llaves, A., and Corcho, O. (2014). Efficient RDF Interchange (ERI) Format for RDF Data Streams. In Mika, P., Tudorache, T., Bernstein, A., Welty, C., Knoblock, C., Vrandečić, D., Groth, P., Noy, N., Janowicz, K., and Goble, C., editors, *The Semantic Web – ISWC 2014*, volume 8797 of *Lecture Notes in Computer Science*, pages 244–259. Springer International Publishing.

Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine.

Finkelstein, L. (1982). Theory and Philosophy of Measurement. In Sydenham, P. H., editor, *Handbook of Measurement Science, Volume 1, Theoretical Fundamentals*, pages 1–30. John Wiley & Sons.

Fiorini, S. R., Abel, M., and Scherer, C. M. (2013). An approach for grounding ontologies in raw data using foundational ontology. *Information Systems*, 38(5):784–799.

Fu, T.-c. (2011). A review on time series data mining. *Engineering Applications of Artificial Intelligence*, 24(1):164–181.

Furno, D., Loia, V., and Veniero, M. (2010). A fuzzy cognitive situation awareness for airport security. *Control and Cybernetics*, 39(4):959–982.

Furno, D., Loia, V., Veniero, M., Anisetti, M., Bellandi, V., Ceravolo, P., and Damiani, E. (2011). Towards an Agent-based Architecture for Managing Uncertainty in Situation Awareness. In *Intelligent Agent (IA), 2011 IEEE Symposium on*, pages 1–6.

Gaber, M. M., Zaslavsky, A., and Krishnaswamy, S. (2005). Mining Data Streams: A Review. *SIGMOD Rec.*, 34(2):18–26.

Gaglio, S., Gatani, L., Lo Re, G., and Ortolani, M. (2007). Understanding the Environment Through Wireless Sensor Networks. In *Proceedings of the 10th Congress of the Italian Association for Artificial Intelligence on AI*IA 2007: Artificial Intelligence and Human-Oriented Computing*, AI*IA '07, pages 72–83, Berlin, Heidelberg. Springer-Verlag.

Galton, A. (2006). On What Goes On: The Ontology of Processes and Events. In Bennett, B. and Fellbaum, C., editors, *Formal Ontology in Information Systems: Proceedings of the Fourth International Conference (FOIS 2006)*, pages 4–11. IOS Press.

Galton, A. and Mizoguchi, R. (2009). The Water Falls but the Waterfall Does Not Fall: New Perspectives on Objects, Processes and Events. *Applied Ontology*, 4(2):71–107.

Gangemi, A. (2005). Ontology Design Patterns for Semantic Web Content. In Gil, Y., Motta, E., Benjamins, R. V., and Musen, M. A., editors, *The Semantic Web – ISWC 2005*, volume 3729 of *Lecture Notes in Computer Science*, pages 262–276. Springer Berlin Heidelberg.

Gangemi, A. and Mika, P. (2003). Understanding the Semantic Web through Descriptions and Situations. In Meersman, R., Tari, Z., and Schmidt, D. C., editors, *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*, volume 2888 of *Lecture Notes in Computer Science*, pages 689–706. Springer Berlin Heidelberg.

Ganguly, A. R., Omitaomu, O. A., Fang, Y., Khan, S., and Bhaduri, B. (2007). Knowledge Discovery from Sensor Data For Scientific Applications. In Gama, J. and Gaber, M. M., editors, *Learning from Data Streams*, pages 205–229. Springer Berlin Heidelberg.

Ganz, F., Barnaghi, P., and Carrez, F. (2013). Information Abstraction for Heterogeneous Real World Internet Data. *Sensors Journal, IEEE*, 13(10):3793–3805.

Ganz, F., Barnaghi, P., and Carrez, F. (2014). Automated Semantic Knowledge Acquisition From Sensor Data. *Systems Journal, IEEE*. (In Press).

Gärdenfors, P. (2004). *Conceptual Spaces: The Geometry of Thought*. MIT press.

Genesereth, M. R. and Nilsson, N. J. (1987). *Logical Foundations of Artificial Intelligence*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Gil, Y. and Miles, S. (2013). PROV Model Primer. W3C Working Group Note, W3C.

Gil, Y., Ratnakar, V., Kim, J., Moody, J., Deelman, E., Gonzalez-Calero, P., and Groth, P. (2011). Wings: Intelligent Workflow-Based Design of Computational Experiments. *Intelligent Systems, IEEE*, 26(1):62–72.

Gorrepati, R. R., Ali, S., and Kim, D.-H. (2013). Hierarchical semantic information modeling and ontology for bird ecology. *Cluster Computing*, 16(4):779–786.

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220.

Guarino, N. and Giaretta, P. (1995). Ontologies and Knowledge Bases: Towards a Terminological Clarification. In Mars, N., editor, *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing*, pages 25–32, Amsterdam. IOS Press.

Guarino, N., Oberle, D., and Staab, S. (2009). What Is an Ontology? In Staab, S. and Studer, R., editors, *Handbook on Ontologies*, International Handbooks on Information Systems, pages 1–17. Springer Berlin Heidelberg.

Guisan, A. and Zimmermann, N. E. (2000). Predictive habitat distribution models in ecology. *Ecological Modelling*, 135(2-3):147–186.

Gurău, C. and Nüchter, A. (2013). Challenges in Using Semantic Knowledge for 3D Object Classification. In Ragni, M., Raschke, M., and Stolzenburg, F., editors, *Proceedings of the KI 2013 Workshop on Visual and Spatial Cognition*, volume 1055 of *KIK - KI & Kognition Workshop Series*, pages 29–35, Koblenz, Germany. CEUR-WS.

Hall, D. and Llinas, J. (1997). An Introduction to Multisensor Data Fusion. *Proceedings of the IEEE*, 85(1):6–23.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The WEKA Data Mining Software: An Update. *SIGKDD Explor. Newsl.*, 11(1):10–18.

Hamed, A., Joutsensaari, J., Mikkonen, S., Sogacheva, L., Dal Maso, M., Kulmala, M., Cavalli, F., Fuzzi, S., Facchini, M., Decesari, S., et al. (2007). Nucleation and growth of new particles in Po Valley, Italy. *Atmospheric Chemistry and Physics*, 7(2):355–376.

Hari, P. and Kulmala, M. (2005). Station for Measuring Ecosystem-Atmosphere Relations (SMEAR II). *Boreal Environment Research*, 10:315–322.

Harnad, S. (1990). The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1-3):335–346.

Harris, S. and Seaborne, A. (2013). SPARQL 1.1 Query Language. Recommendation, W3C.

Hart, J. K. and Martinez, K. (2006). Environmental Sensor Networks: A revolution in the earth system science? *Earth-Science Reviews*, 78(3-4):177–191.

Hasan, S., Curry, E., Banduk, M., and O'Riain, S. (2011). Toward Situation Awareness for the Semantic Sensor Web: Complex Event Processing with Dynamic Linked Data Enrichment. In *Semantic Sensor Network Workshop of the 10th International Semantic Web Conference*, pages 60–72, Bonn, Germany.

Haykin, S. (1998). *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition.

Heflin, J. and Hendler, J. (2001). A Portrait of the Semantic Web in Action. *Intelligent Systems, IEEE*, 16(2):54–59.

Heintz, F., Kvarnström, J., and Doherty, P. (2010). Bridging the sense-reasoning gap: DyKnow – Stream-based middleware for knowledge processing. *Advanced Engineering Informatics*, 24(1):14–26. Informatics for cognitive robots.

Hendler, J. (2001). Agents and the Semantic Web. *Intelligent Systems, IEEE*, 16(2):30–37.

Henson, C., Sheth, A., and Thirunarayan, K. (2012). Semantic Perception: Converting Sensory Observations to Abstractions. *Internet Computing, IEEE*, 16(2):26–34.

Henson, C. A., Pschorr, J. K., Sheth, A. P., and Thirunarayan, K. (2009). SemSOS: Semantic Sensor Observation Service. In *Proceedings of the 2009 International Symposium on Collaborative Technologies and Systems (CTS 2009)*, Baltimore, MD.

Herbin, S., Champagnat, F., Israel, J., Janez, F., Le Saux, B., Leung, V., and Michel, A. (2012). Scene Understanding from Aerospace Sensors: What can be Expected? *AerospaceLab*, 4.

Hill, D. J., Liu, Y., Marini, L., Kooper, R., Rodriguez, A., Futrelle, J., Minsker, B. S., Myers, J., and McLaren, T. (2011). A virtual sensor system for user-generated, real-time environmental data products. *Environmental Modelling & Software*, 26(12):1710–1724.

Hobbs, J. R. and Pan, F. (2006). Time Ontology in OWL. Working draft, W3C.

Horsburgh, J. S., Tarboton, D. G., Maidment, D. R., and Zaslavsky, I. (2011). Components of an environmental observatory information system. *Computers & Geosciences*, 37(2):207–218.

Horsburgh, J. S., Tarboton, D. G., Piasecki, M., Maidment, D. R., Zaslavsky, I., Valentine, D., and Whitenack, T. (2009). An integrated system for publishing environmental observations data. *Environmental Modelling & Software*, 24(8):879–888.

Huang, V. and Javed, M. (2008). Semantic Sensor Information Description and Processing. In *Sensor Technologies and Applications, 2008. SENSORCOMM '08. Second International Conference on*, pages 456–461.

Jajaga, E., Ahmedi, L., and Bexheti, L. A. (2013). Semantic Web Trends on Reasoning Over Sensor Data. In *Proceeding of the 8th South East European Doctoral Student Conference*.

Jakobson, G., Buford, J., and Lewis, L. (2006). A Framework of Cognitive Situation Modeling and Recognition. In *Military Communications Conference, 2006. MILCOM 2006. IEEE*, pages 1–7.

Janowicz, K. (2012). Observation-Driven Geo-Ontology Engineering. *Transactions in GIS*, 16(3):351–374.

Janowicz, K. and Compton, M. (2010). The Stimulus-Sensor-Observation Ontology Design Pattern and its Integration into the Semantic Sensor Network Ontology. In Taylor, K., Ayyagari, A., and Roure, D. D., editors, *Proceedings of the 3rd International Workshop on Semantic Sensor Networks*, volume 668, Shanghai, China. CEUR-WS.

Janowicz, K., van Harmelen, F., Hendler, J. A., and Hitzler, P. (2015). Why the Data Train Needs Semantic Rails. *AI Magazine*, 36(1):5–14.

Keller, M., Schimel, D. S., Hargrove, W. W., and Hoffman, F. M. (2008). A continental strategy for the National Ecological Observatory Network. *Frontiers in Ecology and the Environment*, 6(5):282–284.

Kessler, C., Raubal, M., and Wosniok, C. (2009). Semantic Rules for Context-Aware Geographical Information Retrieval. In Barnaghi, P., Moessner, K., Presser, M., and Meissner, S., editors, *Smart Sensing and Context*, volume 5741 of *Lecture Notes in Computer Science*, pages 77–92. Springer Berlin Heidelberg.

Kimani, S., Lodi, S., Catarci, T., Santucci, G., and Sartori, C. (2004). VidaMine: a visual data mining environment. *Journal of Visual Languages & Computing*, 15(1):37–67.

Klyne, G. and Carroll, J. J. (2004). Resource Description Framework (RDF): Concepts and Abstract Syntax. Recommendation, W3C.

Kokar, M. and Endsley, M. (2012). Situation Awareness and Cognitive Modeling. *Intelligent Systems, IEEE*, 27(3):91–96.

Kokar, M. M., Matheus, C. J., and Baclawski, K. (2009). Ontology-based situation awareness. *Inf. Fusion*, 10(1):83–98.

Koperski, K., Adhikary, J., and Han, J. (1996). Spatial Data Mining: Progress and Challenges. In *Proceedings of the ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery*, pages 1–10, Montreal, Canada.

Kowalski, R. and Sergot, M. (1986). A Logic-based Calculus of Events. *New Generation Computing*, 4(1):67–95.

Kratz, T. K., Arzberger, P., Benson, B. J., Chiu, C.-Y., Chiu, K., Ding, L., Fountain, T., Hamilton, D., Hanson, P. C., Hu, Y. H., Lin, F.-P., McMullen, D. F., Tilak, S., and Wu, C. (2006). Toward a Global Lake Ecological Observatory Network. *Publications of the Karelian Institute*, 145:51–63.

Ladwig, G. and Harth, A. (2011). CumulusRDF: Linked Data Management on Nested Key-Value Stores. In *Proceedings of the 7th International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS2011) at the 10th International Semantic Web Conference (ISWC2011)*.

Lane, N. D. and Georgiev, P. (2015). Can Deep Learning Revolutionize Mobile Sensing? In *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*, Santa Fe, New Mexico.

Lassila, O. and Swick, R. R. (1999). Resource Description Framework (RDF) Model and Syntax Specification. Recommendation, W3C.

Le-Phuoc, D., Quoc, H. N. M., Parreira, J. X., and Hauswirth, M. (2011). The Linked Sensor Middleware—Connecting the real world and the Semantic Web. In *Proceedings of the Semantic Web Challenge at the 10th International Semantic Web Conference*, Bonn, Germany.

Lebo, T., Sahoo, S., and McGuinness, D. (2013). PROV-O: The PROV Ontology. W3C Recommendation, W3C.

Lefort, L., Bobruk, J., Haller, A., Taylor, K., and Woolf, A. (2012). A Linked Sensor Data Cube for a 100 Year Homogenised Daily Temperature Dataset. In Henson, C., Taylor, K., and Corcho, O., editors, *Proceedings of the 5th International Workshop on Semantic Sensor Networks*, volume 904, pages 1–16, Boston, Massachusetts. CEUR-WS.

Leskinen, A., Portin, H., Komppula, M., Miettinen, P., Arola, A., Lihavainen, H., Hatakka, J., Laaksonen, A., and Lehtinen, K. E. J. (2009). Overview of the research activities and results at Puijo semi-urban measurement station. *Boreal Environment Research*, 14:576–590.

Levesque, H., Pirri, F., and Reiter, R. (1998). Foundations for the Situation Calculus. *Linköping Electronic Articles in Computer and Information Science*, 3(18).

Lewis, M., Cameron, D., Xie, S., and Arpinar, I. B. (2006). ES3N: A Semantic Approach to Data Management in Sensor Networks. In *Proceedings of the 1th International Workshop on Semantic Sensor Networks 2006 (SSN2006)*.

Liao, T. W. (2005). Clustering of time series data—a survey. *Pattern Recognition*, 38(11):1857–1874.

Lin, J., Keogh, E., Lonardi, S., and Chiu, B. (2003). A Symbolic Representation of Time Series, with Implications for Streaming Algorithms. In *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, DMKD '03, pages 2–11, New York, NY, USA. ACM.

Liu, J. and Zhao, F. (2005). Towards Semantic Services for Sensor-Rich Information Systems. In *Broadband Networks, 2005. BroadNets 2005. 2nd International Conference on*, volume 2, pages 967–974.

Llaves, A. and Kuhn, W. (2014). An event abstraction layer for the integration of geosensor data. *International Journal of Geographical Information Science*, 28(5):1085–1106.

Loke, S. W. (2004). Representing and reasoning with situations for context-aware pervasive computing: a logic programming perspective. *The Knowledge Engineering Review*, 19:213–233.

Loutfi, A., Coradeschi, S., Daoutis, M., and Melchert, J. (2008). Using Knowledge Representation for Perceptual Anchoring in a Robotic System. *International Journal on Artificial Intelligence Tools*, 17(5):925–944.

Lovett, G. M., Burns, D. A., Driscoll, C. T., Jenkins, J. C., Mitchell, M. J., Rustad, L., Shanley, J. B., Likens, G. E., and Haeuber, R. (2007). Who needs environmental monitoring? *Frontiers in Ecology and the Environment*, 5(5):253–260.

Luckham, D. C. (2002). *The Power of Events*, volume 204. Addison-Wesley Reading.

Ludäscher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger, E., Jones, M., Lee, E. A., Tao, J., and Zhao, Y. (2006). Scientific workflow management and the Kepler system. *Concurrency and Computation: Practice and Experience*, 18(10):1039–1065.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press.

Madden, S. and Franklin, M. (2002). Fjording the Stream: An Architecture for Queries over Streaming Sensor Data. In *Proceedings of the 18th International Conference on Data Engineering*, ICDE '02, pages 555–566, Washington, DC, USA. IEEE Computer Society.

Madden, S., Franklin, M. J., Hellerstein, J. M., and Hong, W. (2003). The Design of an Acquisitional Query Processor for Sensor Networks. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, SIGMOD '03, pages 491–502, New York, NY, USA. ACM.

Madden, S. R., Franklin, M. J., Hellerstein, J. M., and Hong, W. (2005). TinyDB: an acquisitional query processing system for sensor networks. *ACM Trans. Database Syst.*, 30:122–173.

Maedche, A. and Staab, S. (2001). Ontology learning for the Semantic Web. *Intelligent Systems, IEEE*, 16(2):72–79.

Mahmood, A., Shi, K., Khatoon, S., and Xiao, M. (2013). Data Mining Techniques for Wireless Sensor Networks: A Survey. *International Journal of Distributed Sensor Networks*, 2013:1–24.

Mainwaring, A., Culler, D., Polastre, J., Szewczyk, R., and Anderson, J. (2002). Wireless Sensor Networks for Habitat Monitoring. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, WSNA '02, pages 88–97, New York, NY, USA. ACM.

Marascu, A., Pompey, P., Bouillet, E., Wurst, M., Verscheure, O., Grund, M., and Cudre-Mauroux, P. (2014). TRISTAN: Real-time analytics on massive time series using sparse dictionary compression. In *Big Data (Big Data), 2014 IEEE International Conference on*, pages 291–300.

Margara, A., Urbani, J., van Harmelen, F., and Bal, H. (2014). Streaming the Web: Reasoning over dynamic data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 25:24–44.

Martinez, K., Hart, J. K., and Ong, R. (2004). Environmental Sensor Networks. *Computer*, 37(8):50–56.

Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A., and Schneider, L. (2002). WonderWeb Deliverable D17: The WonderWeb Library of Foundational Ontologies.

Maurelli, F., Saigol, Z., and Lane, D. (2014). Cognitive knowledge representation under uncertainty for autonomous underwater vehicles. In *ICRA 2014 Workshop on Persistent Autonomy for Marine Robotics*, Hong Kong, China.

McCarthy, J. and Hayes, P. (1969). Some philosophical problems from the standpoint of artificial intelligence. In Meltzer, B. and Michie, D., editors, *Proceedings of the Fourth Annual Machine Intelligence Workshop*, volume 4 of *Machine Intelligence*. Edinburgh University Press.

Meijers, E. (1986). Defining confusions – Confusing definitions. *Environmental Monitoring and Assessment*, 7(2):157–159.

Mennis, J. and Peuquet, D. J. (2003). The Role of Knowledge Representation in Geographic Knowledge Discovery: A Case Study. *Transactions in GIS*, 7(3):371–391.

Michener, W. K. and Jones, M. B. (2012). Ecoinformatics: supporting ecology as a data-intensive science. *Trends in Ecology & Evolution*, 27:85–93.

Michener, W. K., Porter, J., Servilla, M., and Vanderbilt, K. (2011). Long term ecological research and information management. *Ecological Informatics*, 6(1):13–24.

Miller, R. and Shanahan, M. (2002). Some Alternative Formulations of the Event Calculus. In Kakas, A. C. and Sadri, F., editors, *Computational Logic: Logic Programming and Beyond*, volume 2408 of *Lecture Notes in Computer Science*, pages 452–490. Springer Berlin Heidelberg.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition.

Moodley, D. and Tapamo, J. R. (2011). A Semantic Infrastructure for a Knowledge Driven Sensor Web. In Taylor, K., Ayyagari, A., and Roure, D. D., editors, *Proceedings of the 4th International Workshop on Semantic Sensor Networks*, volume 839, pages 39–54, Bonn, Germany. CEUR-WS.

Moraru, A. and Mladenić, D. (2012). A Framework for Semantic Enrichment of Sensor Data. *Journal of Computing and Information Technology*, 20(3):167–173.

Motik, B., Patel-Schneider, P. F., and Parsia, B. (2012). OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (Second Edition). Recommendation, W3C.

Moumen, A., Oulidi, H. J., Agadi, M., Nehmadou, M., Ben-Daoud, M., Barich, A., Mridekh, A., Mansouri, B. E., Boutaleb, S., Mohammed, K. B. H., Essahlaoui, A., and Eljaafari, S. (2014). A Sensor Web for Real-Time Groundwater Data Monitoring in Morocco. *Journal of Geographic Information System*, 6(6):613–623.

Müller, H., Cabral, L., Morshed, A., and Shu, Y. (2013). From RESTful to SPARQL: A Case Study on Generating Semantic Sensor Data. In Corcho, O., Henson, C., and Barnaghi, P., editors, *Proceedings of the 6th International Workshop on Semantic Sensor Networks*, volume 1063, pages 51–66, Sydney, Australia. CEUR-WS.

Mulligan, M. and Wainwright, J. (2004). Modelling and Model Building. In Wainwright, J. and Mulligan, M., editors, *Environmental Modelling: Finding Simplicity in Complexity*, chapter Modelling and Model Building, pages 7–73. John Wiley & Sons, Ltd.

Myers, T. S. and Trevathan, J. (2013). Semantic Support for Hypothesis-Based Research from Smart Environment Monitoring and Analysis Technologies. *International Journal of Computer, Information, Systems and Control Engineering*, 7(8):505–513.

Neches, R., Fikes, R., Finin, T., Gruber, T., Patil, R., Senator, T., and Swartout, W. R. (1991). Enabling Technology for Knowledge Sharing. *AI Magazine*, 12(3):36–56.

Negru, S. (2012). SemaKoDE: Hybrid System for Knowledge Discovery in Sensor-Based Smart Environments. In Brambilla, M., Tokuda, T., and Tolksdorf, R., editors, *Web Engineering*, volume 7387 of *Lecture Notes in Computer Science*, pages 448–451. Springer Berlin Heidelberg.

Nittel, S. (2009). A Survey of Geosensor Networks: Advances in Dynamic Environmental Monitoring. *Sensors*, 9(7):5664–5678.

Nittel, S., Labrinidis, A., and Stefanidis, A. (2008). Introduction to Advances in Geosensor Networks. In Nittel, S., Labrinidis, A., and Stefanidis, A., editors, *GeoSensor Networks*, volume 4540 of *Lecture Notes in Computer Science*, pages 1–6. Springer Berlin Heidelberg.

Padovitz, A., Loke, S. W., Zaslavsky, A., and Burg, B. (2004). Towards A General Approach for Reasoning about Context, Situations and Uncertainty in Ubiquitous Sensing: Putting Geometrical Intuitions to Work. In *2nd International Symposium on Ubiquitous Computing Systems (UCS'04)*, Tokyo, Japan.

Pantelopoulos, A. and Bourbakis, N. (2010). A Survey on Wearable Sensor-Based Systems for Health Monitoring and Prognosis. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 40(1):1–12.

Parkinson, K. J., Day, W., and Leach, J. E. (1980). A Portable System for Measuring the Photosynthesis and Transpiration of Graminaceous Leaves. *Journal of Experimental Botany*, 31(5):1441–1453.

Perron, P. (2014). Space Weather Situational Awareness and Its Effects upon a Joint, Interagency, Domestic, and Arctic Environment. *Canadian Military Journal*, 14(4).

Perry, M. and Herring, J. (2012). OGC GeoSPARQL - A Geographic Query Language for RDF Data. Technical Report OGC 11-052r4, Open Geospatial Consortium Inc.

Persson, A., Coradeschi, S., Rajasekaran, B., Krishna, V., Loutfi, A., and Alirezaie, M. (2013). I Would Like Some Food: Anchoring Objects to Semantic Web Information in Human-Robot Dialogue Interactions. In Herrmann, G., Pearson, M. J., Lenz, A., Bremner, P., Spiers, A., and Leonards, U., editors, *Social Robotics*, volume 8239 of *Lecture Notes in Computer Science*, pages 361–370. Springer International Publishing.

Peters, D. P. C., Havstad, K. M., Cushing, J., Tweedie, C., Fuentes, O., and Villanueva-Rosales, N. (2014). Harnessing the power of big data: infusing the scientific method with machine learning to transform ecology. *Ecosphere*, 5(6):1–15.

Pongpaichet, S., Singh, V. K., Gao, M., and Jain, R. (2013). EventShop: Recognizing Situations in Web Data Streams. In *Proceedings of the 22nd International Conference on World Wide Web Companion*, WWW '13 Companion, pages 1359–1368, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

Porter, J., Arzberger, P., Braun, H.-W., Bryant, P., Gage, S., Hansen, T., Hanson, P., Lin, C.-C., Lin, F.-P., Kratz, T., Michener, W., Shapiro, S., and Williams, T. (2005). Wireless Sensor Networks for Ecology. *BioScience*, 55(7):561–572.

Probst, F. (2006). Ontological Analysis of Observations and Measurements. In Raubal, M., Miller, H., Frank, A., and Goodchild, M., editors, *Geographic Information Science*, volume 4197 of *Lecture Notes in Computer Science*, pages 304–320. Springer Berlin Heidelberg.

Prud'hommeaux, E. and Carothers, G. (2014). RDF 1.1 Turtle: Terse RDF Triple Language. Recommendation, W3C.

Prud'hommeaux, E. and Seaborne, A. (2008). SPARQL Query Language for RDF. Recommendation, W3C.

Rabiner, L. and Juang, B. (1986). An Introduction to Hidden Markov Models. *ASSP Magazine, IEEE*, 3(1):4–16.

Randell, D., Cui, Z., and Cohn, A. (1992). A Spatial Logic based on Regions and Connections. In Nebel, B., Rich, C., and Swartout, W., editors, *Principles of Knowledge Representation and Reasoning*, pages 165–176, San Mateo, CA, USA. Morgan Kaufmann.

Raskino, M., Fenn, J., and Linden, A. (2005). Extracting Value From the Massively Connected World of 2015. Research G00125949, Gartner, Inc.

Reggia, J. A. and Peng, Y. (1987). Modeling diagnostic reasoning: a summary of parsimonious covering theory. *Computer Methods and Programs in Biomedicine*, 25(2):125–134.

Remagnino, P. and Foresti, G. (2005). Ambient Intelligence: A New Multidisciplinary Paradigm. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 35(1):1–6.

Resch, B., Schmidt, D., and Blaschke, T. (2007). Enabling Geographic Situational Awareness in Emergency Management. In *Proceedings of the 2nd Geospatial Integration for Public Safety Conference*, pages 15–17, New Orleans, Louisiana, US.

Rettinger, A., Lösch, U., Tresp, V., d'Amato, C., and Fanizzi, N. (2012). Mining the Semantic Web. *Data Mining and Knowledge Discovery*, 24(3):613–662.

Rew, R. and Davis, G. (1990). NetCDF: an interface for scientific data access. *Computer Graphics and Applications, IEEE*, 10(4):76–82.

Riker, W. H. (1957). Events and Situations. *The Journal of Philosophy*, 54(3):57–70.

Robson, B. J. (2014). When do aquatic systems models provide useful predictions, what is changing, and what is next? *Environmental Modelling & Software*, 61:287–296.

Roda, F. and Musulin, E. (2014). An ontology-based framework to support intelligent data analysis of sensor measurements. *Expert Systems with Applications*, 41(17):7914–7926.

Roddick, J. F., Hornsby, K., and Spiliopoulou, M. (2001). An Updated Bibliography of Temporal, Spatial, and Spatio-temporal Data Mining Research. In Roddick, J. F. and Hornsby, K., editors, *Temporal, Spatial, and Spatio-Temporal Data Mining*, volume 2007 of *Lecture Notes in Computer Science*, pages 147–163. Springer Berlin Heidelberg.

Rundel, P. W., Graham, E. A., Allen, M. F., Fisher, J. C., and Harmon, T. C. (2009). Environmental sensor networks in ecological research. *New Phytologist*, 182(3):589–607.

Sagl, G., Resch, B., Mittlboeck, M., Hochwimmer, B., and Lippautz, M. (2012). Standardised geosensor webs and web-based geo-processing for near real-time situational awareness in emergency management . *Int. J. Business Continuity and Risk Management*, 3(4):339–358.

Salfinger, A., Neidhart, D., Retschitzegger, W., Schwinger, W., and Mitsch, S. (2014). SEM^2 Suite — Towards a Tool Suite for Supporting Knowledge Management in Situation Awareness Systems. In Joshi, J., Bertino, E., Thuraisingham, B., and Liu, L., editors, *Proceedings of the 15th IEEE International Conference on Information Reuse and Integration*, pages 351–360, San Francisco, CA, USA. IEEE Systems, Man, and Cybernetics Society (SMC).

Salmon, P. M., Stanton, N. A., Jenkins, D. P., Walker, G. H., Young, M. S., and Aujla, A. (2007). What Really Is Going on? Review, Critique and Extension of Situation Awareness Theory. In Harris, D., editor, *Engineering Psychology and Cognitive Ergonomics*, volume 4562 of *Lecture Notes in Computer Science*, pages 407–416. Springer Berlin Heidelberg.

Salmon, P. M., Stanton, N. A., and Young, K. L. (2012). Situation awareness on the road: review, theoretical and methodological issues, and future directions. *Theoretical Issues in Ergonomics Science*, 13(4):472–492.

Sarma, S., Venkatasubramanian, N., and Dutt, N. (2014). Sense-making from Distributed and Mobile Sensing Data: A Middleware Perspective. In *Proceedings of the 51st Annual Design Automation Conference*, DAC '14, pages 68:1–68:6, New York, NY, USA. ACM.

Shanahan, M. (2005). Perception as Abduction: Turning Sensor Data Into Meaningful Representation. *Cognitive Science*, 29(1):103–134.

Sharpe, P. (1990). Forest modeling approaches: compromises between generality and precision. In Dixon, R. K., Meldahl, R. S., Ruark, G. A., and Warren, W. G., editors, *Process Modeling of Forest Growth Responses to Environmental Stress*, pages 180–190. Timber Press, Portland, OR.

Sheth, A., Henson, C., and Sahoo, S. (2008). Semantic Sensor Web. *Internet Computing, IEEE*, 12(4):78–83.

Sirin, E., Parsia, B., Grau, B., Kalyanpur, A., and Katz, Y. (2007). Pellet: A practical OWL-DL reasoner. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(2):51–53.

Solomatine, D., See, L., and Abrahart, R. (2008). Data-Driven Modelling: Concepts, Approaches and Experiences. In Abrahart, R. J., See, L. M., and Solomatine, D. P., editors, *Practical Hydroinformatics*, volume 68 of *Water Science and Technology Library*, pages 17–30. Springer Berlin Heidelberg.

Stanton, N. A., Salmon, P. M., Walker, G. H., and Jenkins, D. P. (2010). Is situation awareness all in the mind? *Theoretical Issues in Ergonomics Science*, 11(1-2):29–40.

Stanton, N. A., Stewart, R., Harris, D., Houghton, R. J., Baber, C., McMaster, R., Salmon, P., Hoyle, G., Walker, G., Young, M. S., Linsell, M., Dymott, R., and Green, D. (2006). Distributed situation awareness in dynamic systems: theoretical development and application of an ergonomics methodology. *Ergonomics*, 49(12-13):1288–1311. PMID: 17008257.

Stewart Hornsby, K. and King, K. (2008). Linking Geosensor Network Data and Ontologies to Support Transportation Modeling. In Nittel, S., Labrinidis, A., and Stefanidis, A., editors, *GeoSensor Networks*, volume 4540 of *Lecture Notes in Computer Science*, pages 191–209. Springer Berlin Heidelberg.

Stocker, M., Baranizadeh, E., Hamed, A., Rönkkö, M., Virtanen, A., Laaksonen, A., Portin, H., Komppula, M., and Kolehmainen, M. (2013). Acquisition and Representation of Knowledge for

Atmospheric New Particle Formation. In Hřebíček, J., Schimak, G., Kubásek, M., and Rizzoli, A. E., editors, *Environmental Software Systems. Fostering Information Sharing*, volume 413 of *IFIP Advances in Information and Communication Technology*, pages 98–108. Springer Berlin Heidelberg.

Stocker, M., Kauhanen, O., Hiirsalmi, M., Saarela, J., Rossi, P., Rönkkö, M., Hytönen, H., Kotovirta, V., and Kolehmainen, M. (2015a). A Software System for the Discovery of Situations Involving Drivers in Storms. In Denzer, R., Argent, R. M., Schimak, G., and Hřebíček, J., editors, *Environmental Software Systems. Infrastructures, Services and Applications*, volume 448 of *IFIP Advances in Information and Communication Technology*, pages 226–234. Springer International Publishing.

Stocker, M., Rönkkö, M., and Kolehmainen, M. (2012a). Making Sense of Sensor Data Using Ontology: A Discussion for Residential Building Monitoring. In Iliadis, L., Maglogiannis, I., Papadopoulos, H., Karatzas, K., and Sioutas, S., editors, *Artificial Intelligence Applications and Innovations*, volume 382 of *IFIP Advances in Information and Communication Technology*, pages 341–350. Springer Boston.

Stocker, M., Rönkkö, M., and Kolehmainen, M. (2012b). Making Sense of Sensor Data Using Ontology: A Discussion for Road Vehicle Classification. In Seppelt, R., Voinov, A., Lange, S., and Bankamp, D., editors, *2012 International Congress on Environmental Modelling and Software*, pages 2387–2394, Leipzig, Germany. International Environmental Modelling & Software Society.

Stocker, M., Rönkkö, M., and Kolehmainen, M. (2014). Towards an Ontology for Situation Assessment in Environmental Monitoring. In Ames, D. P., Quinn, N. W., and Rizzoli, A. E., editors, *Proceedings of the 7th International Congress on Environmental Modelling and Software*, volume 3, pages 1281–1288, San Diego, California, USA. International Environmental Modelling & Software Society.

Stocker, M., Rönkkö, M., and Kolehmainen, M. (2015b). Provenance in Systems for Situation Awareness in Environmental Monitoring. In Denzer, R., Argent, R. M., Schimak, G., and Hřebíček, J., editors, *Environmental Software Systems. Infrastructures, Services and Applications*, volume 448 of *IFIP Advances in Information and Communication Technology*, pages 169–177. Springer International Publishing.

Stocker, M., Rönkkö, M., Villa, F., and Kolehmainen, M. (2011). The Relevance of Measurement Data in Environmental Ontology Learning. In Hřebíček, J., Schimak, G., and Denzer, R., editors, *Environmental Software Systems. Frameworks of eEnvironment*, volume 359 of *IFIP Advances in Information and Communication Technology*, pages 445–453. Springer Berlin Heidelberg.

Stocker, M., Shurpali, N., Taylor, K., Burba, G., Rönkkö, M., and Kolehmainen, M. (2015c). Emrooz: A Scalable Database for SSN Observations. In Kyzirakos, K., Henson, C., Perry, M., Varanka, D., and Grütter, R., editors, *Joint Proceedings of the 1st Joint International Workshop on Semantic Sensor Networks and Terra Cognita (SSN-TC 2015) and the 4th International Workshop on Ordering and Reasoning (OrdRing 2015) co-located with the 14th International Semantic Web Conference (ISWC 2015)*, volume 1488, pages 1–12, Bethlehem, PA, USA. CEUR-WS.

Studer, R., Benjamins, V., and Fensel, D. (1998). Knowledge engineering: Principles and methods. *Data & Knowledge Engineering*, 25(1–2):161–197.

Swartout, B., Patil, R., Knight, K., and Russ, T. (1996). Toward Distributed Use of Large-Scale Ontologies. In *Procedings of the 10th Knowledge Acquisition for Knowledge-Based Systems Workshop*, Banff, Canada.

Swartout, W. and Tate, A. (1999). Ontologies. *Intelligent Systems and their Applications, IEEE*, 14(1):18–19.

Tamea, G., Cusmai, M., Palo, A., Priscoli, F., and Cimmino, A. (2014). Situation awareness in airport environment based on Semantic Web technologies. In *Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), 2014 IEEE International Inter-Disciplinary Conference on*, pages 174–180.

Taylor, K. and Leidinger, L. (2011). Ontology-Driven Complex Event Processing in Heterogeneous Sensor Networks. In Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., Leenheer, P., and Pan, J., editors, *The Semanic Web: Research and Applications*, volume 6644 of *Lecture Notes in Computer Science*, pages 285–299. Springer Berlin Heidelberg.

Taylor, P. (2014). OGC WaterML 2.0: Part 1 – Timeseries. OGC Implementation Standard - Corrigendum OGC 10-126r4, Open Geospatial Consortium Inc.

Thessler, S., Kooistra, L., Teye, F., Huitu, H., and Bregt, A. K. (2011). Geosensors to Support Crop Production: Current Applications and User Requirements. *Sensors*, 11(7):6656–6684.

Tolle, G., Polastre, J., Szewczyk, R., Culler, D., Turner, N., Tu, K., Burgess, S., Dawson, T., Buonadonna, P., Gay, D., and Hong, W. (2005). A Macroscope in the Redwoods. In *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems*, SenSys '05, pages 51–63, New York, NY, USA. ACM.

Tollefson, J. (2011). US launches eco-network. *Nature*, 476(135).

Uddling, J., Hogg, A. J., Teclaw, R. M., Carroll, M. A., and Ellsworth, D. S. (2010). Stomatal uptake of O_3 in aspen and aspen-birch forests under free-air CO_2 and O_3 enrichment. *Environmental Pollution*, 158(6):2023–2031.

Velikova, M., Novák, P., Huijbrechts, B., Laarhuis, J., Hoeksma, J., and Michels, S. (2014). An Integrated Reconfigurable System for Maritime Situational Awareness. In Schaub, T., Friedrich, G., and O'Sullivan, B., editors, *Proceedings of the 21st European Conference on Artificial Intelligence*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 1197–1202, Prague, Czech Republic. IOS Press.

Wang, P., Zheng, J. G., Fu, L., Patton, E. W., Lebo, T., Ding, L., Liu, Q., Luciano, J. S., and McGuinness, D. L. (2011). A Semantic Portal for Next Generation Monitoring Systems. In Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., and Blomqvist, E.,

editors, *The Semantic Web – ISWC 2011*, volume 7032 of *Lecture Notes in Computer Science*, pages 253–268. Springer Berlin Heidelberg.

Wang, W., Barnaghi, P., Cassar, G., Ganz, F., and Navaratnam, P. (2012). Semantic Sensor Service Networks. In *Sensors, 2012 IEEE*, pages 1–4.

Wei, W. and Barnaghi, P. (2009). Semantic Annotation and Reasoning for Sensor Data. In Barnaghi, P., Moessner, K., Presser, M., and Meissner, S., editors, *Smart Sensing and Context*, volume 5741 of *Lecture Notes in Computer Science*, pages 66–76. Springer Berlin Heidelberg.

Weiss, C., Karras, P., and Bernstein, A. (2008). Hexastore: Sextuple Indexing for Semantic Web Data Management. *Proc. VLDB Endow.*, 1(1):1008–1019.

Wetz, P., Trinh, T.-D., Do, B.-L., Anjomshoaa, A., Kiesling, E., and Tjoa, A. M. (2014). Towards an Environmental Information System for Semantic Stream Data. In *Proceedings of the 28th International Conference on Informatics for Environmental Protection*, pages 637–644, Oldenburg, Germany. BIS-Verlag.

Whitehouse, K., Zhao, F., and Liu, J. (2006). Semantic Streams: A Framework for Composable Semantic Interpretation of Sensor Data. In Römer, K., Karl, H., and Mattern, F., editors, *Wireless Sensor Networks*, volume 3868 of *Lecture Notes in Computer Science*, pages 5–20. Springer Berlin Heidelberg.

Worboys, M. (2005). Event-oriented approaches to geographic phenomena. *International Journal of Geographical Information Science*, 19(1):1–28.

Worboys, M. and Hornsby, K. (2004). From Objects to Events: GEM, the Geospatial Event Model. In Egenhofer, M. J., Freksa, C., and Miller, H. J., editors, *Geographic Information Science*, volume 3234 of *Lecture Notes in Computer Science*, pages 327–343. Springer Berlin Heidelberg.

Wu, L. (2012). Representing and Inferring Events from Deforestation Observations. In Gensel, J., Josselin, D., and Vandenbroucke, D., editors, *Proceedings of the AGILE'2012 International Conference on Geographic Information Science*, pages 80/392–85/392, Avignon, France.

Xiao, F., Shea, G. Y. K., Wong, M. S., and Campbell, J. (2014). An automated and integrated framework for dust storm detection based on OGC web processing services. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-2:151–156.

Yao, Y. and Gehrke, J. (2002). The Cougar Approach to In-network Query Processing in Sensor Networks. *SIGMOD Rec.*, 31(3):9–18.

Ye, J., Dobson, S., and McKeever, S. (2012). Situation identification techniques in pervasive computing: A review. *Pervasive and Mobile Computing*, 8(1):36–66.

Ye, J., Stevenson, G., and Dobson, S. (2014). USMART: An Unsupervised Semantic Mining Activity Recognition Technique. *ACM Trans. Interact. Intell. Syst.*, 4(4):16:1–16:27.

Yu, J., Davis, P., Gould, S., and Taylor, K. (2014). Linked Data Approach For Automated Failure Detection In Pressure Sewers Using Real-Time Sensor Data. In *Proceedings of the 11th International Conference on Hydroinformatics*, New York City, USA.

Yu, J. and Taylor, K. (2013). Event dashboard: Capturing user-defined semantics events for event detection over real-time sensor data. In Corcho, O., Henson, C., and Barnaghi, P., editors, *Proceedings of the 6th International Workshop on Semantic Sensor Networks*, volume 1063, pages 19–34, Sydney, Australia. CEUR-WS.

Zhang, D., Sullivan, T., Briciu-Burghina, C., Murphy, K., McGuinness, K., O'Connor, N. E., Smeaton, A., and Regan, F. (2014). Detection and Classification of Anomalous Events in Water Quality Datasets Within a Smart City-Smart Bay Project. *International Journal on Advances in Intelligent Systems*, 7(1&2):167–178.

Markus Stocker

Situation Awareness in Environmental Monitoring

Environmental monitoring data contribute to advancing our understanding of natural and human-made systems. Monitoring data are increasingly often voluminous sensor data. To turn data into actionable knowledge, software systems need to integrate advanced techniques in data processing, information acquisition, and knowledge representation. For case studies in intelligent transportation systems, atmospheric science, and agricultural science this dissertation proposes to model observed phenomena as objects in situations and discusses the representation and processing of situational knowledge acquired from data in situation-aware environmental monitoring systems.

UNIVERSITY OF
EASTERN FINLAND

Publications of the University of Eastern Finland
Dissertations in Forestry and Natural Sciences

ISBN: 978-952-61-1907-6 (PRINT)
ISBN: 978-952-61-1908-3 (PDF)
ISSN: 1798-5668 (PRINT)
ISSN: 1798-5676 (PDF)