

Abstractions from Sensor Data with Complex Event Processing and Machine Learning

Markus Stocker, Mauno Rönkkö, Mikko Kolehmainen

University of Eastern Finland, P.O. Box 1627, 70211 Kuopio, Finland
markus.stocker@uef.fi, mauno.ronkko@uef.fi, mikko.kolehmainen@uef.fi

Abstract: Environmental knowledge systems that build on sensor-based environmental monitoring rely on techniques in knowledge acquisition and representation to interpret the numbers obtained in measurement for what they tell about the monitored environment. Languages and systems in knowledge representation and reasoning, specifically Semantic Web technologies, support the formulation and execution of rules, a technique that enables deductive inference in a knowledge base. This technique has been used to demonstrate inference on sensor data. While the approach certainly has its merits, it is often demonstrated for numerical thresholds and, thus, for relatively trivial “semantic enrichment.” In reality, knowledge acquisition tasks of interest to environmental knowledge systems that build on sensor-based environmental monitoring are often more challenging. They rely on advanced computational techniques and models, e.g. in machine learning or complex event processing. In order to ease the formulation and execution of such tasks, systems need to integrate such techniques. Towards this end, we present the integration of machine learning with WEKA and complex event processing with Esper in Wavellite.

Keywords: sensor data; knowledge acquisition; machine learning; complex event processing; wavellite.

1 INTRODUCTION

Wavellite¹ (Stocker et al., 2014b, 2013a, 2014a, 2013b) is a modelling and software framework for situation awareness (Endsley, 1995) in environmental monitoring. It builds on (typically, but not exclusively) sensor-based environmental monitoring and supports the representation of situational knowledge acquired, or extracted, from sensor data. Hence, its core functionality is for situation assessment (Endsley, 1995), i.e. for *gaining* situation awareness (Salfinger et al., 2013). Wavellite also builds on knowledge representation and reasoning, in particular ontology languages (W3C OWL Working Group, 2012; Brickley et al., 2004) and, more generally, Semantic Web (Berners-Lee et al., 2001) technologies.

The logical structure of the Wavellite architecture consists of four layers: measurement, observation, derivation, and situation. The four layers build on each other, from measurement to situation. Each layer serves a purpose and abstracts from underlying complexity. Layers consist of components. We briefly describe the layers with their most important components. For the interested reader, Stocker et al. (2014a) includes a diagram representing the logical structure of the Wavellite architecture.

The measurement layer abstracts from the physical sensor network and data communication links and protocols. It consists of measurement readers. A measurement reader implements the software logic required to process sensor data into measurement results. In addition to contextual data, measurement results include measurement values, i.e. the numbers assigned to properties of objects or events of the real world in the process of measurement (Finkelstein, 1982).

¹<http://www.uef.fi/en/envi/projects/wavellite>

The observation layer abstracts from the heterogeneity of sensor data. It includes observation engines, which translate measurement results into sensor observations. The semantics of sensor observations are defined by the Semantic Sensor Network (SSN) ontology (Compton et al., 2012). This translation overcomes the syntactic and semantic heterogeneity of sensor data by aligning sensor data to the syntax and semantics of the SSN ontology.

The derivation layer abstracts from the sensor network dimension of sensor observations. It includes dataset engines and derivation engines. Dataset engines translate sensor observations into dataset observations, elements of datasets. The semantics of dataset observations, and datasets, are defined by the RDF Data Cube Vocabulary (QB) (Cyganiak et al., 2013). Derivation engines perform computations on dataset observations. Such computation amounts to dataset processing, in that the dataset observations of one or more source datasets are processed to dataset observations of a target dataset. Computations may be for, e.g., despiking, aggregation, interpolation, merging, filtering.

The situation layer abstracts from data with situational knowledge for the monitored environment. It includes situation engines, which extract situational knowledge from datasets. Computations may include methods in, e.g., machine learning or complex event processing. Situation engines represent situational knowledge as situations. The semantics of situations are defined by the Situation Theory Ontology (STO) (Kokar et al., 2009) which is grounded in Situation Theory (Barwise and Perry, 1983; Devlin, 1995). There are two aspects to situations. First, a situation is “a structured part of Reality” (Devlin, 1995). Thus, a parent observing a child constructing a castle in a sandbox is a situation. Second, a situation is an ontological individual instance of the ontology class for situations, defined by the STO. In order to represent situations and information about them that is known to be true or false, situations are formalized by means of the expression $s \models \sigma$, meaning that the infon σ is “made factual” by the situation s . An infon σ has the structure $\langle\langle R, a_1, \dots, a_m, i \rangle\rangle$, whereby R is an n -place relation and a_1, \dots, a_m ($m \leq n$) are objects appropriate for the argument places i_1, \dots, i_m of R , and $i = 0, 1$ is the polarity, i.e. the ‘truth value’ of the infon. If $i = 1$ then the objects a_1, \dots, a_m stand in the relation R ; else the objects do not stand in the relation R . Objects a_1, \dots, a_m may be, among other types, relevant individuals (e.g. the child, the parent, or the sandbox) and temporal or spatial locations.

In the context of environmental monitoring and science, awareness is for situations, i.e. structured parts of reality (Devlin, 1995), monitored by environmental sensor networks (Martinez et al., 2004; Hart and Martinez, 2006). Relevant individuals in situations are, typically, environmental phenomena, such as storm, wind, or pollution. Relevant individuals may also be non-physical, such as growing season. Generally, environmental sensor networks do not monitor situations directly. Instead, they monitor the properties of certain phenomena. It is in situation assessment that situational knowledge is extracted from data obtained in monitoring the properties of phenomena. For instance, we may gain awareness of the situation whereby the 2014 Finnish growing seasons is longer than average by situation assessment on data obtained via an environmental sensor network that monitors the temperature of air.

As suggested by the survey conducted by Salfinger et al. (2013), systems for situation awareness have been developed and demonstrated predominantly for the military and safety & security domains. This is even though much of the generic architecture of situation awareness systems described by Salfinger et al. (2013, Fig. 1) is applicable to the domain of environmental monitoring and science. The main difference seems to be the client and consumer of represented situational knowledge, which in the latter domain is, typically, a researcher, rather than an operator of a control centre. As an example, Stocker et al. (2013a, 2014a) demonstrated the application of Wavellite to the acquisition and representation of situational knowledge for atmospheric phenomena, specifically new particle formation, an environmental phenomenon studied by aerosol scientists.

The use of ontology-based systems for the management of, and reasoning on, sensor data has recently gained some popularity, not least due to the development of the SSN ontology. Beyond the mere representation and, thus, management of sensor observations, ontology-based systems may leverage on inference engines in order to automatically represent knowledge that is implicit to sensor observations and a domain model (Sheth et al., 2008; Henson et al., 2009; Wei and Barnaghi, 2009; Kessler et al., 2009; Stocker et al., 2011). It has, however, been recognized that such deductive inference is only one of many techniques aimed at “making sense of sensor data” and, specifically, the problem of converting sensory observations to abstractions reflecting the sensed environment (Henson et al., 2012). Hence the predominant development of hybrid systems in concrete applications (Jajaga et al., 2013). Inherent to systems for situation awareness,

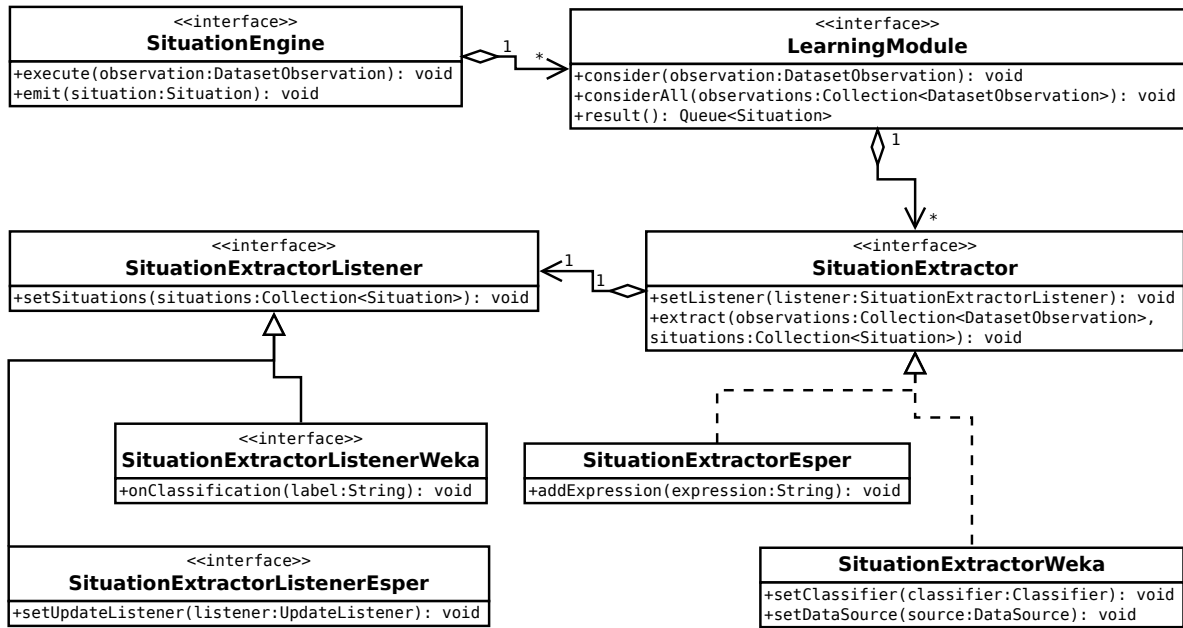


Figure 1. Diagram showing the Wavellite interfaces and classes that are involved in situational knowledge extraction from dataset observations and knowledge representation as situations, using machine learning classification and complex event processing.

this problem is of interest to various domains, including the *Internet of Things* (Ganz et al., 2013) or robotics and autonomous systems (Daoutis, 2013). Various approaches for tackling aspects of this problem have been discussed in the literature (e.g. Gaglio et al. (2007); Henson et al. (2012); Barnaghi et al. (2012); Calbimonte et al. (2012); Stocker et al. (2014b)).

However, to the best of our knowledge, a generic approach that suits arbitrary monitored environments and is specific to environmental monitoring and science does not exist. The domain has traditionally relied on a broad range of methods, encompassing, among others, digital signal processing, complex event processing, machine learning, and physically-based modelling. With Wavellite we aim at a coherent representation of data involved in, as well as the integration of methods relevant to, the representation of (situational) knowledge acquired from sensor data. Towards this aim, in this paper, we discuss the integration of machine learning classification and complex event processing in Wavellite, specifically using the libraries WEKA (Hall et al., 2009) and Esper.² WEKA implements a set of machine learning algorithms and is a popular software for data mining. Esper implements an engine for event stream processing and supports the analysis of streams for complex events expressed using an event processing language. Both WEKA and Esper are open source, are implemented in Java, and can serve as libraries in applications.

2 INTEGRATION

In Wavellite, situation engines orchestrate knowledge extraction from datasets and knowledge representation as situations. The actual work required for knowledge extraction is executed by learning modules, to which situation engines associate. Specifically, in executing a dataset observation, a situation engine needs to decide which learning module is going to consider the dataset observation. In addition, a situation engine emits the situations returned by learning modules, i.e. implements the logic for what happens to represented situations (e.g. a request for storing). Figure 1 provides an overview of the Wavellite interfaces that are involved in situational knowledge extraction and representation. For the interested reader, Stocker et al. (2014a) includes a diagram that places the situation engine in context of the entire Wavellite architecture.

²<http://esper.codehaus.org/>

Generally, the representation of a situation can require on one or more knowledge extraction tasks. Due to the vast diversity of situational knowledge of interest in environmental monitoring (and, thus, the vast diversity of knowledge extraction tasks) situation engines and learning modules are typically domain specific and, thus, require domain implementations. This diversity also motivates the integration of various computational methods in, e.g., complex event processing, machine learning, or physically-based modelling. Therefore, in order to support applications in their implementation of knowledge extraction tasks from dataset observations, Wavellite wraps the heterogeneous families of computational methods behind Wavellite operators, the so-called situation extractors. Learning modules can associate to one or more situation extractors.

Situation extractors are implemented for specific libraries of computational methods, such as Esper or WEKA. Naturally, a Wavellite application can implement own situation extractors, for instance one backed by a specific physically-based model. Extractors operate on dataset observations provided by learning modules. Generally, the learning module configures situation extractors according to application requirements, e.g. with a particular Event Processing Language (EPL) expression in case of the extractor backed by Esper or with a particular machine learning classifier and labelled dataset in case of the extractor backed by WEKA. Moreover, learning modules provide extractors with listeners, which are called upon completed extraction, e.g. upon classification by WEKA or upon events detected by Esper. Listeners implement how extracted situational knowledge is represented as situations.

3 EXAMPLES

We briefly describe how the integration works on two concrete examples, one for complex event processing and one for machine learning.

The example for machine learning builds on our previous work (Stocker et al., 2012, 2014b). The setting is as follows. A small sensor network consisting of three vibration sensors is used to monitor the pavement vibration of a road section. The sensors are installed at three, roughly equidistant, points on a metal bar that is horizontally inserted into the ground below the pavement at one side of the road. The purpose of the sensor network, and of road-pavement vibration monitoring, is to detect and classify vehicles as they move on the road section. The presence of a vehicle close to a sensor induces vibration that is distinct from background vibration. With data processing, vibration over time is transformed into vibration patterns indexed in time (e.g. a pattern every second). The data of vibration patterns is represented as dataset observations. Hence, the dataset represents an unbounded set of vibration patterns indexed in time. Given a classifier and appropriately labelled training samples, we develop a learning module that uses the situation extractor backed by WEKA to classify dataset observations, i.e. vibration patterns. The result of such classification indicates the presence (or absence) of a vehicle as well as vehicle characteristics (e.g. light or heavy). Such extracted knowledge is part of situations describing the state of the monitored road section, in particular w.r.t. traffic. Situations are returned by the learning module to the situation engine which eventually forwards situations to other components of the Wavellite architecture.

The example for complex event processing is a proof of concept with implementation included in the wavellite-example project³ and has the following setting. A sensor measures wind speed over time at a certain location. Such measurement data is represented as sensor observations, which are then translated to observations of a unbounded dataset for wind speed data. We implement a learning module that uses the situation extractor backed by Esper to detect situations of “strong wind.” The extractor is configured with a listener, which is called upon events detected by Esper. The listener specifies how Esper event information is translated into situational knowledge. Furthermore, the extractor is configured with an EPL statement that selects dataset observations for wind speed with value greater than 10. Dataset observations are provided to the learning module as input. The learning module forwards dataset observations to the extractor, which translates them to Esper events and submits the events to the Esper runtime. According to the EPL statement, our listener is called when the Esper runtime detects a dataset observation with value for wind speed greater than 10. Upon such detection, the learning module creates a situation of “strong wind” that includes information for wind speed and time. Finally, situations queued by the learning module can be retrieved by the client of the module (e.g. a situation engine) for further processing. This example is, admittedly, trivial. However, it is

³<https://github.com/markusstocker/wavellite-example>

easy to see that, by exposing Esper EPL in the extractor, learning modules can specify more complex EPL statements on dataset observations.

4 DISCUSSION AND CONCLUSION

Based on our work and experience in environmental monitoring for scientific applications we are increasingly confident that (ontology-based) *practical* systems aimed at abstractions from sensor data should support and integrate a broad range of computational methods. This is particularly motivated by the vast diversity of abstractions and the, perhaps more often than not, winding road from data to abstractions.

It may be attractive particularly to ontology-based systems to use inference mechanisms to tackle the problem. While (deductive) reasoning services may be an argument that speak for ontology-based systems, we think they are generally not sufficient in environmental monitoring for scientific applications. Indeed, the application of machine learning to sensor data is often aimed at obtaining abstractions (e.g. Kubat et al. (1998); Athanasiadis et al. (2003); Moraru et al. (2010); Ahmad et al. (2010)). We, thus, think that ontology-based systems aimed at abstractions from sensor data should support and integrate a broad range of computational methods, including machine learning. With Wavellite we aim at such integration and in this paper we briefly discussed the integration of complex event processing and machine learning using two popular open source software packages.

Taylor and Leidinger (2011) present a method that uses ontology to drive complex event detection in sensor data. The ontology models domain knowledge as well as concepts for event, observation, and sensor network. (The authors also build on the SSN ontology.) The ontology drives a user interface that facilitates the definition of events of interest to a domain. Such definitions, represented in terms of the ontology, are transformed to EPL statements (the authors used Coral8, nowadays Sybase CEP). The CEP engine is responsible for processing the statements on (streamed) observation data. While the method presented by Taylor and Leidinger (2011) is far more sophisticated than the current state in Wavellite, the core of the idea and implementation is comparable. Both systems operate on observation data with shared representation grounded in ontology. Also, both systems allow for the definition of events which are processed by a CEP engine. A key difference is that Taylor and Leidinger (2011) ground event definition in ontology and implement a translation of definitions to EPL statements whereas Wavellite, in its current state, requires implementers to write EPL statements. The second difference is that, in addition to complex event processing, Wavellite integrates machine learning. A third difference is that in Wavellite information extracted by means of complex event processing is part of knowledge represented as situations. In contrast, in Taylor and Leidinger (2011) the result of complex event processing is an alert, in form of an email or mobile phone text message. In Wavellite, such alerts could be the result of complex event processing on dataset observations or situations.

On a similar line, Llaves and Kuhn (2014) present the ontology-based architecture and implementation for an event abstraction layer that analyses observation time series to create event streams. The event abstraction ontology proposed by the authors also extends the SSN ontology and introduces the core concept for *event abstraction*. The system accepts time series data encoded in O&M,⁴ which are translated into CEP objects processed by the CEP engine, implemented by Esper. Event patterns are pairs consisting of an Esper EPL statement and an ontological class for the event type. Upon events detected by the CEP engine, an ontological individual, instance of the ontological class associated to the returned event, is created. Individuals are then published to an event bus, implemented by RabbitMQ.⁵ Consumers, such as a triple store, can, thus, subscribe to the bus to further process the data. The core of the idea and implementation is comparable to Taylor and Leidinger (2011) as well as Wavellite. Contrary to Taylor and Leidinger (2011) and Llaves and Kuhn (2014), Wavellite is however not limited to complex event processing.

Our discussion begs the question: events or situations? Is one of them the better concept for abstractions obtained from sensor data? Are they complementary? Having chosen the STO in Wavellite, we are perhaps biased towards situations. However, quoting Barwise and Perry (1980), Devlin (2006) notes that “events and episodes are situations in time, scenes are visually perceived situations, changes are sequences of situations, and facts are situations enriched (or polluted) by language.” This seems to at least suggest that

⁴<http://www.opengeospatial.org/standards/om>

⁵<https://www.rabbitmq.com/>

situations are more generic than events. We also highlight Riker (1957) who called situations “the boundaries of events” and events the action occurring between situations. We do not attempt to discuss this matter further, being it beyond the scope of this paper as well as our expertise. However, one practical advantage of situations over events seems to be that, contrary to Taylor and Leidinger (2011) and Llaves and Kuhn (2014) who constructed two distinct event ontologies, in Wavellite we could simply adopt the STO.

We have briefly described the integration of complex event processing and machine learning in Wavellite, as learning modules of its architecture, using two popular software packages. We conclude this paper by noting that physically-based models that create abstractions from sensor data can arguably be understood as Wavellite learning modules. The architecture does not set restrictions on such models being third party or implemented natively. In fact, we are in the process of developing a Wavellite application in which sensor data and a physically-based model are used to create abstractions for the disease pressure at agricultural fields. In this application, information returned by the disease pressure model is part of situational knowledge representing the state of agricultural fields, in particular w.r.t. disease pressure.

ACKNOWLEDGMENTS

This research is funded by the Academy of Finland project “FResCo: High-quality Measurement Infrastructure for Future Resilient Control Systems” (Grant number 264060).

REFERENCES

- Ahmad, S., Kalra, A., and Stephen, H. (2010). Estimating soil moisture using remote sensing data: A machine learning approach. *Advances in Water Resources*, 33(1):69–80.
- Athanasiadis, I. N., Kaburlasos, V. G., Mitkas, P. A., and Petridis, V. (2003). Applying machine learning techniques on air quality data for real-time decision support. In *First international NAISO symposium on information technologies in environmental engineering (ITEE'2003)*, Gdansk, Poland.
- Barnaghi, P., Ganz, F., Henson, C., and Sheth, A. (2012). Computing Perception from Sensor Data. Technical report, knosis.org.
- 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. (1983). *Situations and attitudes*. Bradford books. MIT Press.
- Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The Semantic Web. *Scientific American*, 284(5):28–37.
- Brickley, D., Guha, R., and McBride, B. (2004). RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation, W3C.
- 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.
- Compton, M., Barnaghi, P., Bermudez, L., Garca-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.
- Cyganiak, R., Reynolds, D., and Tennison, J. (2013). The RDF Data Cube Vocabulary. W3C Candidate Recommendation, W3C.
- Daoutis, M. (2013). Knowledge Based Perceptual Anchoring. *KI - Künstliche Intelligenz*, 27(2):179–182.
- Devlin, K. (1995). *Logic and information*. Cambridge University Press.

- 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.
- Endsley, M. (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.
- Finkelstein, L. (1982). *Theory and Philosophy of Measurement*. John Wiley & Sons.
- 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.
- Ganz, F., Barnaghi, P., and Carrez, F. (2013). Information Abstraction for Heterogeneous Real World Internet Data. *Sensors Journal, IEEE*, 13(10):3793–3805.
- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The WEKA Data Mining Software: An Update. In *SIGKDD Explorations*, volume 11.
- 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.
- 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.
- Jajaga, E., Ahmedi, L., and Bexheti, L. A. (2013). Semantic Web Trends on Reasoning Over Sensor Data. Proceeding of the 8th South East European Doctoral Student Conference.
- 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.
- Kokar, M. M., Matheus, C. J., and Baclawski, K. (2009). Ontology-based situation awareness. *Inf. Fusion*, 10(1):83–98.
- Kubat, M., Holte, R., and Matwin, S. (1998). Machine Learning for the Detection of Oil Spills in Satellite Radar Images. *Machine Learning*, 30(2-3):195–215.
- Llaves, A. and Kuhn, W. (2014). An event abstraction layer for the integration of geosensor data. *International Journal of Geographical Information Science*. (In Press).
- Martinez, K., Hart, J. K., and Ong, R. (2004). Environmental Sensor Networks. *Computer*, 37(8):50–56.
- Moraru, A., Pesko, M., Porcius, M., Fortuna, C., and Mladenic, D. (2010). Using machine learning on sensor data. In *Information Technology Interfaces (ITI), 2010 32nd International Conference on*, pages 573–578.
- Riker, W. H. (1957). Events and situations. *The Journal of Philosophy*, 54(3):pp. 57–70.
- Salfinger, A., Retschitzegger, W., and Schwinger, W. (2013). Maintaining Situation Awareness Over Time – A Survey on the Evolution Support of Situation Awareness Systems. In *Conference on Technologies and Applications of Artificial Intelligence (TAAI 2013)*, pages 274–281. IEEE Computer Society, IEEE.
- Sheth, A., Henson, C., and Sahoo, S. (2008). Semantic Sensor Web. *Internet Computing, IEEE*, 12(4):78–83.
- Stocker, M., Baranizadeh, E., Hamed, A., Rönkkö, M., Virtanen, A., Laaksonen, A., Portin, H., Komppula, M., and Kolehmainen, M. (2013a). Acquisition and Representation of Knowledge for Atmospheric New Particle Formation. In Hřebíček, J., Schimak, G., Kubsek, 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., Baranizadeh, E., Portin, H., Komppula, M., Rönkkö, M., Hamed, A., Virtanen, A., Lehtinen, K., Laaksonen, A., and Kolehmainen, M. (2014a). Representing situational knowledge acquired from sensor data for atmospheric phenomena. *Environmental Modelling & Software*, 58:27–47.
- Stocker, M., Rönkkö, M., and Kolehmainen, M. (2012). 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. iEMSS.
- Stocker, M., Rönkkö, M., and Kolehmainen, M. (2013b). The Wavellite Modelling and Software Framework for Situation Awareness in Environmental Monitoring. *Environmental Monitoring and Assessment*. (Submitted).
- Stocker, M., Rönkkö, M., and Kolehmainen, M. (2014b). Situational knowledge representation for traffic observed by a pavement vibration sensor network. *IEEE Transactions on Intelligent Transportation Systems*. (In Press).
- Stocker, M., Rönkkö, M., Villa, F., and Kolehmainen, M. (2011). The Relevance of Measurement Data in Environmental Ontology Learning. In *Environmental Software Systems. Frameworks of eEnvironment*, volume 359 of *IFIP Advances in Information and Communication Technology*, pages 445–453. Springer Boston.
- 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 Semantic Web: Research and Applications*, volume 6644 of *Lecture Notes in Computer Science*, pages 285–299. Springer Berlin Heidelberg.
- W3C OWL Working Group (2012). OWL 2 Web Ontology Language. W3C Recommendation, W3C.
- 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.