Situational Knowledge Representation for Traffic Observed by a Pavement Vibration Sensor Network

Markus Stocker, Mauno Rönkkö, and Mikko Kolehmainen

***Abstract*—Information systems that build on sensor networks often process data produced by measuring physical properties. These data can serve in the acquisition of knowledge for real-world situations that are of interest to information services and, ultimately, to people. Such systems face a common challenge, namely the considerable gap between the data produced by measurement and the abstract terminology used to describe real-world situations. We present and discuss the architecture of a software system that utilizes sensor data, digital signal processing, machine learning, and knowledge representation and reasoning to acquire, represent, and infer knowledge about real-world situations observable by a sensor network. We demonstrate the application of the system to vehicle detection and classification by measurement of road pavement vibration. Thus, real-world situations involve vehicles and information for their type, speed, and driving direction.**

***Index Terms*—Knowledge acquisition, knowledge representation, machine learning, sensor data, sensor networks, traffic monitoring.**

I. Introduction

We propose a software system architecture and implementation for the continuous and automated representation of knowledge for real-world situations observable by a sensor network. In this paper, we demonstrate the application of the software system to intelligent transportation systems. Thus, real-world situations involve vehicles and information for their type, speed, and driving direction.

According to Finkelstein [1], "measurement is 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." A sensor is a device that performs measurement, in that it transforms the signal of a physical property (e.g., heat) into numbers or, more generally, into data [2]. Sensor measurement is, hence, the process of recurrent application of such transformation for certain temporal and spatial locations. The result of sensor measurement is sensor data. Sensor data represent the change of the signal over time.

Despite recent advancements in sensor data management, processing, and query [2]–[4], as well as semantic description of sensors and data [5]–[7], making sense of sensor data is an ongoing challenge [8]–[10] because of the difference in the degree to which sensor data represents information about a signal and information about, or related to, a physical property [11]. In other words, it is a challenge because of the considerable gap between data produced by measurement and abstract terminology [12] used by people to describe (the properties of) real-world objects or events.

We are interested in *situations* involving real-world objects that affect a physical property, for which a signal is measured by means of sensors. In this paper, vehicles are the real-world objects and road pavement vibration is the physical property. We present the architecture of a software system that utilizes digital signal processing, machine learning, and knowledge representation and reasoning to acquire, represent, and infer knowledge about real-world situations involving vehicles. The system aims at reducing the gap between road pavement vibration measurement data and abstract terminology used to describe real-world situations involving vehicles.

Digital signal processing techniques are iteratively applied to a sliding window over sensor data to enhance the vibration signal and to transform sensor data (time domain) into patterns (frequency domain). Machine learning is used to classify patterns. We employ multilayer perceptron (MLP) feedforward artificial neural networks [13]. Techniques in knowledge representation are utilized to formally represent domain concepts, instances, and relations. A concept of interest to our domain is the vibration sensor. The (installed) sensors are represented as instances of this concept. An instance may have a number of relations, e.g., to a spatial location. We represent sensors and observations using the Semantic Sensor Network Ontology (SSNO) [14].[1] SSNO is an "ontology for describing the capabilities of sensors, the act of sensing and the resulting observations" [15]. We employ the Situation Theory Ontology[2] (STO) [16] to represent knowledge about real-world situations, which are acquired from observations. The STO captures the key aspects of the situation theory developed by Barwise and Perry [17] and extended by Devlin [18]. The theory relates to the work on situation awareness by Endsley [19], [20] as it encompasses most of the concepts discussed in [16]. Both the SSNO and the STO serve as upper ontologies from which we extend to accommodate domain knowledge. The hybrid use of the SSNO and the STO allows for a multilevel abstraction of sensor measurement data and the use of appropriate terminology and formalization at each level.

Manuscript received April 12, 2013; revised August 16, 2013 and November 20, 2013; accepted December 22, 2013. Date of publication February 4, 2014; date of current version August 1, 2014. The infrastructure to access and collect vibration and camera data, as well as the data, are part of research funded by Tekes, the Finnish Funding Agency for Technology and Innovation (funding decision number 40075/09). The Associate Editor for this paper was P. Grisleri.

The authors are with the Department of Environmental Science, University of Eastern Finland, 70211 Kuopio, Finland (e-mail: markus.stocker@uef.fi; mauno.ronkko@uef.fi; mikko.kolehmainen@uef.fi).

Digital Object Identifier 10.1109/TITS.2013.2296697

[1] http://purl.oclc.org/NET/ssnx/ssn
[2] http://vistology.com/ont/2008/STO/STO.owl

Thus, observations are semantically enriched measurements, and situations are acquired from observations.

The main contribution of this paper lies in a thorough analysis of ontology-based situational knowledge representation for traffic observed by a sensor network. Knowledge is acquired from processed sensor data using machine learning. This paper builds on a software system aimed at near real-time representation of situational knowledge acquired from sensor data. The proposed system is generic enough to serve different domains. We present its application to intelligent transportation systems.

II. Materials and Methods

Here, we first present the materials used in this paper, namely the sensor network, the retrieved data, and software. We then detail the methods utilized to process sensor data, as well as to acquire, represent, and infer knowledge.

A. Materials

Road pavement vibration was measured using three CEF C3M01 accelerometer vibration sensors developed by Control Express Finland (CEF) Oy[3] for condition monitoring and machinery maintenance. (CEF C3M01 sensors are currently manufactured by Webrosensor Oy[4] as WBS CM301.) The sensor network was installed at the training site of the Finnish Emergency Services College, Kuopio, Finland. The site is used for emergency response training in simulated situations involving, for instance, vehicles or buildings that are on fire. The area can be accessed by vehicle, and its paved light traveled roads are for different types of vehicles, such as ambulances and fire trucks. The three accelerometer vibration sensors were part of a wider sensor network that consisted of chemical sensors, weather stations, acoustic sensors, and surveillance cameras. The sensor network was installed and maintained for a Finnish research project that aimed at the development of systems for the monitoring of an operational environment.

The accelerometer vibration sensors—hereafter referred to as sensing devices sd_1, sd_2, and sd_3—were installed with a relative distance of approximately 45 m at the right side (with respect to the surveillance camera, described later) along one of the roads at the training site. Each sensor was mounted on a metal bar that penetrated approximately 1 m into the ground, roughly 0.5 m below the paved road surface. The sensors measured ground vibration, including vibration induced by vehicles. We visually monitored the road using an AXIS 211W Wireless Network Camera with an Outdoor Antenna Kit AXIS 211W [21]. The camera was positioned on top of a viewpoint tower located nearby the road and directed toward the monitored road section.

We retrieved vibration data from the three CEF C3M01 sensors (sampling frequency of 2 kHz) and image data from the AXIS camera (1–3 frames/s) for a total of 6 h on August 30, 2011, between 10:00 A.M. and 4:00 P.M. We retrieved 42 962 432 values from sd_1, 42 937 345 from sd_2, and 42 988 810 from sd_3. We retrieved 25 076 image files from the AXIS camera. Data retrieved from CEF C3M01 sensors were persistently stored using Apache Cassandra.[5] Files retrieved from the AXIS camera were stored individually to a disk using the acquisition time, in milliseconds, as file name.

We developed a software system to automatically process *historical* vibration data stored in Apache Cassandra and to detect vibration patterns. Data for vibration patterns formed training data sets, aimed at supervised machine learning. We used WEKA[6] [22] (version 3.6.5) to train and evaluate MLP feedforward artificial neural network classifiers. Situational knowledge acquired from vibration data was represented in a domain ontology, using the Web Ontology Language (OWL) [23] and Resource Description Framework (RDF) [24] knowledge representation languages. We used Protégé[7] (version 4.1) and Apache Jena [25][8] (version 2.7.0) to manually and programmatically manage the ontology. To demonstrate rule-based inference, we defined two domain rules and evaluated them over represented situations. We implemented and evaluated the rules manually as processes executed over represented situations. We also tested the Semantic Web Rule Language (SWRL) [26]. The work was performed on a workstation with an Intel Core 2 3.16-GHz CPU with 8-GB RAM and an Ubuntu 11.04 Linux operating system. The software was written in Java.

We developed a software system framework for *continuous and (near) real-time* processing of sensor data to acquire and represent situational knowledge. The system builds on Storm,[9] which is a distributed real-time computation system. The architecture consists of three layers: measurement, observation, and situation. Fig. 1 provides a schematic overview. Each layer consists of components. Components correspond to Storm nodes and may associate to modules. Modules provide services such as storage, inference, machine learning, complex event processing, and digital signal processing. Components communicate via Storm streams. Components and their communication links thus form a Storm topology. A Storm topology can be executed on a single machine or a cluster.

At the measurement layer, a measurement engine is an abstraction for sensors. Its responsibility is to implement the necessary software logic to acquire data from sensors and process them into measurements. Measurements are data objects with an associated timestamp. Data objects may be of primitive type, such as numbers, or complex type, such as data structures. Measurements are then forwarded to streams. At the observation layer, an observation engine subscribes to measurement streams. An observation engine is responsible for the semantic enrichment of measurements. Such semantic enrichment occurs according to the SSNO. Observations are then forwarded to streams. An observation store may exist at the observation layer and subscribe to observation streams. An observation store is responsible for storing incoming observations to the knowledge store. At the situation layer, a situation engine subscribes to observation streams. A situation engine is responsible for

[3]http://www.cef.fi/
[4]http://www.wbs.fi/
[5]http://cassandra.apache.org
[6]http://www.cs.waikato.ac.nz/ml/weka/
[7]http://protege.stanford.edu
[8]http://incubator.apache.org/jena/
[9]http://storm-project.net/

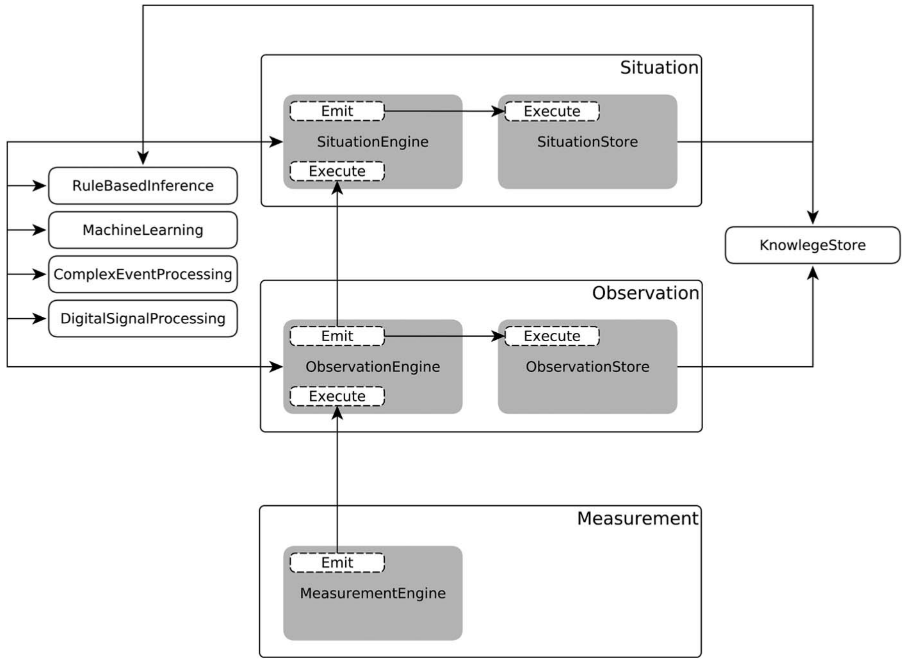

Fig. 1. Logical structure of the system architecture for (near) real-time representation of situational knowledge acquired from sensor data showing the three layers of measurement, observation, and situation, as well as the main components and modules, and their interactions.

the implementation of situational knowledge acquisition tasks from observations. To do so, the situation engine orchestrates services provided by various modules, e.g., for digital signal processing, complex event processing, machine learning, or inference. A situation engine represents knowledge for situations according to the STO. Situations are then forwarded to streams. A situation store may subscribe to situation streams and store situations to the knowledge store.

B. Methods

Digital signal processing operations implemented by a sliding window over persisted sensor data were employed to iteratively process vibration data into vibration patterns. Specifically, a bandpass filter was applied to enhance the vibration signal *possibly* induced by vehicles and a fast Fourier transform was applied to transform filtered sensor data into a vibration pattern. Vibration patterns possibly corresponding to vehicle occurrences were automatically identified and manually labeled. For manual labeling, we first processed the camera data to visually identify vehicle occurrences. Each identified vehicle occurrence was described with metadata for the vehicle type (e.g., a personal car), the time at which the vehicle crossed the approximate location of sd_2 and the driving side of the vehicle. The metadata were used to link occurrences (manually) identified in camera data with those (automatically) detected in vibration data. By linking, we thus constructed labeled data, which were used to generate training data sets for supervised machine learning. Machine learning was for the two tasks of vehicle detection (training classes `vehicle` and `no-vehicle`) and vehicle classification (training classes `light` and `heavy`). Training data sets only included samples occurring between 10:00 A.M. and 2:50 P.M., such that we may discuss the results of our simulated real-time experiments for occurrences that were unseen by algorithms during the training phase. Classification performance was evaluated using tenfold cross validation, meaning that the training data set was partitioned into ten disjoint and equal-sized folds, and for each fold, a classifier was trained using the other nine folds and then tested on the fold. Intermediate results were averaged. For both the vehicle detection and vehicle classification tasks, the MLP networks consisted of ten input neurons, two output neurons, and one hidden layer with six hidden neurons. The learning rate and momentum were set to 0.3 and 0.2, respectively.

The situation theory developed by Devlin [18] formalizes the semantics of situations by means of the expression $s \models \sigma$ (read "s supports σ") meaning that the infon σ is "made factual" by the situation s. According to the definition by Devlin, the object $\ll R, a_i, \ldots, a_m, i \gg$ is a well-defined infon if R is an n-place relation and $a_1, \ldots, a_m$ $(m \leq n)$ are objects appropriate for the argument places $i_1, \ldots, i_m$ of R, and if the filling of argument places $i_1, \ldots, i_m$ is sufficient to satisfy the minimality conditions for R, and $i = 0$, 1 is the polarity. Minimality conditions "determine which particular groups of argument roles need to be filled in order to produce an infon" [18]. The polarity is the "truth value" of the infon. If $i = 1$, then the objects $a_1, \ldots, a_m$ stand in the relation R; else, the objects do not stand in the relation R. Parameters, which are denoted $\dot{a}$, make reference to *arbitrary* objects of a given type. For instance, $\dot{l}$ and $\dot{t}$ typically denote parameters for arbitrary objects of type spatial location and temporal location, respectively. Anchors are a mechanism to assign values to parameters. Hence, parameter $\dot{t}$ may anchor the value for the current time.

In order to represent situational knowledge for vehicles and information about their type, speed, and driving direction, we created a domain ontology that imported both the SSNO and the STO. It extended the SSNO and the STO to accommodate knowledge about observations and situations that are specific to the domain. Fig. 3 provides an overview of the axioms used. We extended the SSNO to accommodate the class `Accelerometer` for domain-specific sensing devices, and stated that sd_1, sd_2, and sd_3 are individual sensing devices, instances of `Accelerometer`. The individual *pavement* represents the monitored road section and is the domain-specific SSNO feature of interest, i.e., the real-world object (physical entity). The individual *vibration* represents the monitored property of the SSNO feature of interest. Hence, the individual accelerometers sd_1, sd_2, and sd_3 observe *vibration*, which is a property of *pavement*. We extended the STO to accommodate the class `Vehicle` as a domain-specific class of STO relevant individuals. Sensing devices, hence accelerometers, are also STO relevant individuals. STO relevant individuals are objects in infons. The classes `LightVehicle` and `HeavyVehicle` are subclasses of `Vehicle`. We also stated the disjointness of `LightVehicle` and `HeavyVehicle`, meaning that a vehicle can be either light or heavy but not both. Finally, the individuals *near*, *driving-side*, and *driving-speed* are domain-specific STO relations.

We simulated a real-time context by programmatically processing vibration data between 2:50 P.M. and 3:00 P.M. on August 30, 2011, for the sensors sd_i. Measurements were first translated to SSNO observations. We did not store observations but, rather, processed them directly to acquire knowledge for situations. A situation was for a vehicle being *near* an accelerometer sensing device. Formally, the situation s supports the infon σ, i.e., $s \models \sigma$, whereby σ is the *near*-relation infon $\ll near, \dot{\psi}, \dot{sd}, \dot{l}, \dot{t}, 1 \gg$ with parameters $\dot{\psi}$, $\dot{sd}$, $\dot{l}$, and $\dot{t}$ that may be anchored to a specific vehicle, accelerometer sensing device, spatial location, and temporal location, respectively. Hence, upon detection of vehicles in observations by accelerometer sd_i, we populated the ontology with individuals, ω, instances of STO `Situation`, with STO relation `supportedInfon` to an individual, σ, instance of STO `ElementaryInfon`. Infons with *near*-relation anchor two STO relevant individuals, a `Vehicle` and an `Accelerometer`, as well as STO attributes for spatial and temporal locations. Hence, such infons state that, at temporal location t and spatial location l, the vehicle ψ was near the accelerometer sensing device sd_i. Given situations $s \models \ll near, \dot{\psi}, \dot{\mathrm{sd}}, \dot{l}, \dot{t}, 1 \gg$ for detected vehicles, we classified individuals $\texttt{Vehicle}(\psi)$ as $\texttt{LightVehicle}(\psi)$ or $\texttt{HeavyVehicle}(\psi)$.

To demonstrate rule inference, we defined two domain rules $p \rightarrow q$, being p and q the rule antecedent and the rule consequent, respectively. The first rule stated that for situations where $s_p \neq s_q$ with *near*-relation infons

$$s_p \models \ll near, \psi_p, \mathrm{sd}_i, l_i, t_p, 1 \gg$$
$$s_q \models \ll near, \psi_q, \mathrm{sd}_j, l_j, t_q, 1 \gg$$

whereby $\mathrm{sd}_i \neq \mathrm{sd}_j$ (hence, $l_i \neq l_j$) and $|t_p - t_q| < 8$ s, the anchored vehicles ψ_p and ψ_q are *same*, i.e., the same physical entity. We represented this relation using the OWL axiom $\texttt{sameAs}(\psi_p, \psi_q)$. This rule was motivated by the relative distance of approximately 45 m between consecutive sensors and the average low-volume traffic.

TABLE I
VEHICLE OCCURRENCES BETWEEN 10:00 A.M. AND 2:50 P.M. AS IDENTIFIED IN CAMERA DATA AND LINKED TO VIBRATION DATA, LISTED ACCORDING TO THE ACTUAL VEHICLE TYPE. LINKING RESULTED IN 87 `vehicle` (TRUE POSITIVE) AND 211 `no-vehicle` (FALSE POSITIVE) DETECTION OCCURENCES FOR sd_1; 134 AND 42, FOR sd_2; AND 133 AND 32, FOR sd_3. ADDITIONALLY, WE CONSIDERED 134 SAMPLES WITHOUT VEHICLE OCCURRENCE FOR EACH SENSING DEVICE (`background`)

Label	Camera	sd_1	sd_2	sd_3
mini-cleaner	7	0	4	5
mini-lifter	3	1	3	3
personal-car	9	3	7	8
van	24	15	20	17
ambulance	15	9	14	14
fire-van	28	10	18	19
pickup-truck	8	2	5	5
truck	42	27	36	34
fire-truck	21	16	20	20
bucket-digger	8	4	7	8
vehicle	165	87	134	133
background		134	134	134
no-vehicle		211	42	32
Total		432	310	299

Furthermore, for situation pairs (s_p, s_q) with $\texttt{sameAs}(\psi_p, \psi_q)$, the second rule inferred the velocity of the vehicle. Velocity determined the vehicle's driving side and speed. We used the STO relations *driving-side* and *driving-speed* to represent this knowledge. Hence, for such pairs (s_p, s_q), we inferred the following infons:

$$s_p \models \ll driving\text{-}side, \psi_p, \nu, 1 \gg$$
$$s_p \models \ll driving\text{-}speed, \psi_p, \eta, 1 \gg$$
$$s_q \models \ll driving\text{-}side, \psi_q, \nu, 1 \gg$$
$$s_q \models \ll driving\text{-}speed, \psi_q, \eta, 1 \gg$$

where ν is an STO attribute for the driving side, and η is an STO attribute for the driving speed. The STO attribute value for driving side is `right`, with respect to the camera perspective, if $l_i < l_j$ and $t_p < t_q$ or if $l_i > l_j$ and $t_p > t_q$, and `left` otherwise. (Note that, with respect to the camera perspective, for sd_1, sd_2, and sd_3, it holds that $l_1 < l_2 < l_3$, meaning that the spatial location l_1 is closest to the camera. We also assume that drivers respect driving rules.) The STO attribute value for driving speed was computed as $0.045 \times (3600/|t_p - t_q|)$ with STO dimensionality [km/h].

III. RESULTS

Table I summarizes the vehicle occurrences manually identified in camera data and automatically detected in vibration data between 10:00 A.M. and 2:50 P.M. We identified 165 vehicles in camera images, of which 87 (53%) were automatically detected by sd_1, 134 (81%) by sd_2, and 133 (81%) by sd_3. In two cases, the three sensors detected (vehicle) vibration, but we could not confirm the presence of the vehicle (and label) due to missing camera data for the corresponding time interval. The label `no-vehicle` was used for detections that did not correspond to vehicle-induced vibration or corresponded to both vehicle-induced vibration and vibration-like signal induced by unexplained factors. As shown in Table I, with 211 `no-vehicle` detection occurrences, sd_1 was considerably more "noisy" than

TABLE II
THE KEY CHARACTERISTICS OF THE DATA SETS USED TO TRAIN CLASSIFIERS FOR THE MACHINE LEARNING VEHICLE DETECTION (VD) TASK WITH TRAINING CLASSES (C) `vehicle` (V) AND `no-vehicle` (NV) AND VEHICLE CLASSIFICATION (VC) TASK WITH TRAINING CLASSES `light` (L) AND `heavy` (H). THE TABLE SHOWS THE MAPPING OF ACTUAL VEHICLE LABELS (L) TO TRAINING CLASSES AND CORRESPONDING NUMBER OF TRAINING SAMPLES, FOR EACH SENSING DEVICE

	Machine learning tasks							
	VD				VC			
L	C	sd_1	sd_2	sd_3	C	sd_1	sd_2	sd_3
mini-cleaner mini-lifter personal-car van ambulance fire-van pickup-truck	V	87	134	133	L	40	71	71
truck fire-truck bucket-digger					H	47	63	62
background no-vehicle	NV	345	176	166				

TABLE III
SUMMARY OF PRECISION (P) AND RECALL (R) FIGURES FOR THE CLASSES (C) `no-vehicle` (NV) AND `vehicle` (V) OF THE VEHICLE DETECTION (VD) TASK AND THE CLASSES `light` (L) AND `heavy` (H) OF THE VEHICLE CLASSIFICATION (VC) TASK FOR THE THREE SENSING DEVICES (SD)

	C	SD	P	R
VD	NV	sd_1	0.967	0.933
		sd_2	0.971	0.943
		sd_3	0.953	0.97
	V	sd_1	0.768	0.874
		sd_2	0.928	0.963
		sd_3	0.962	0.94
VC	H	sd_1	0.83	0.83
		sd_2	0.721	0.778
		sd_3	0.842	0.774
	L	sd_1	0.8	0.8
		sd_2	0.788	0.732
		sd_3	0.816	0.873

either sd_2 or sd_3. Indeed, many detection occurrences in sd_1 were labeled as `no-vehicle` because we detected noise or because we detected signal but the corresponding window included also noise. We do not know the reason for such noise or for its relative abundance in sd_1 compared with sd_2 or sd_3. Given the comparable installation of the three sensors, the difference may be due to malfunction or manufacturing issues.

Table II summarizes, for the three sensors, the key characteristics of the data sets used to train classifiers for the machine learning vehicle detection and classification tasks. The table shows how the actual vehicle labels were mapped to training classes and the corresponding number of training samples per class. Classification performance (correctly classified instances) for the vehicle *detection* task resulted to be 92% for sd_1, 95% for sd_2, and 96% for sd_3. Classification performance for the vehicle *classification* task resulted to be 82% for sd_1, 75% for sd_2, and 83% for sd_3. Table III is a summary of precision and recall figures for the classes `no-vehicle` and `vehicle` of the vehicle detection task and the classes `light` and `heavy` of the vehicle classification task, for the three sensing devices. Notably, the precision of class `vehicle` (vehicle detection task) for sensing device sd_1 is relatively low, which

TABLE IV
SUMMARY OF THE CONFUSION MATRICES FOR THE VEHICLE DETECTION AND VEHICLE CLASSIFICATION TASKS FOR THE THREE SENSING DEVICES. GIVEN THE CONFUSION MATRICES WE CAN CALCULATE THE PRECISION AND RECALL FIGURES

(a) Vehicle detection with classes `no-vehicle` and `vehicle`. For class `no-vehicle` and sensing device sd_1 precision is calculated as $\frac{322}{322+11} = 0.967$ and recall as $\frac{322}{322+23} = 0.933$.

		Predicted					
		no-vehicle			vehicle		
		sd_1	sd_2	sd_3	sd_1	sd_2	sd_3
Actual	no-vehicle	322	166	161	23	10	5
	vehicle	11	5	8	76	129	125

(b) Vehicle classification with classes `heavy` and `light`. For class `light` and sensing device sd_3 precision is calculated as $\frac{62}{62+14} = 0.816$ and recall as $\frac{62}{62+9} = 0.873$.

		Predicted					
		heavy			light		
		sd_1	sd_2	sd_3	sd_1	sd_2	sd_3
Actual	heavy	39	49	48	8	14	14
	light	8	19	9	32	52	62

TABLE V
VEHICLE OCCURRENCES WITH INFORMATION ON RIGHT (R) OR LEFT (L) DRIVING SIDE (D), BETWEEN 2:50 PM AND 3:00 PM AS IDENTIFIED IN CAMERA DATA AND REPRESENTED IN SITUATIONS OF VEHICLES NEAR SENSORS. MACHINE LEARNING CLASSIFICATION (C) FOR LIGHT VEHICLE (LV) OR HEAVY VEHICLE (HV) IS SHOWN. MISCLASSIFICATION IS HIGHLIGHTED

Occurrence			sd_1		sd_2		sd_3	
Label	D	H:m	s	C	s	C	s	C
personal-car	R	14:50	36	Lv	39	Lv	43	**Hv**
van	R	14:53	20	Lv	24	Lv	27	**Hv**
truck	L	14:57	56	Hv	52	Hv	49	Hv
fire-truck	R	14:58	39	Hv	43	**Lv**	46	Hv

reflects the unbalanced data set with 87 versus 345 samples for classes `vehicle` and `no-vehicle`, respectively. Table IV(a) is a summary of the confusion matrices for the vehicle detection task and the three sensing devices, and Table IV(b) is the corresponding summary for the vehicle classification task.

As result of the simulated real-time experiments, Table V summarizes the key elements of the 12 situations for the four vehicles near the three sensors between 2:50 P.M. and 3:00 P.M. In particular, for situations $s \models \ll near,\ \dot{\psi},\ \dot{sd},\ \dot{l},\ \dot{t},\ 1 \gg$, we show the anchored temporal location t (for time; date is August 30, 2011) and the most specific vehicle class of the anchored individual ψ. Infons for driving side and speed, i.e., vehicle velocity, as well as `sameAs` relations among vehicles (not shown), were inferred according to the two rules described earlier. For instance, rule inference correctly inferred that the vehicle near sd_2 at 14:53:24 is `sameAs` the vehicle near sd_3 at 14:53:27 as well as `sameAs` the vehicle near sd_1 at 14:53:20. As we can see, machine learning correctly classified the vehicles 9 times out of 12 (75%).

Fig. 2 depicts the RDF graph describing the situation $s \models \ll near, v, \mathrm{sd}_1,$ "2011-08-30T14:57:56+03:00", $1 \gg$, meaning that at 14:57:56 the (heavy) vehicle v was near the accelerometer sensing device sd_1. Note that, for readability, we do not show the information for the spatial location anchored by infon i, which we model after the spatial location of sd_1. Contrary to represented observations, represented situations were persistently stored by the knowledge store.

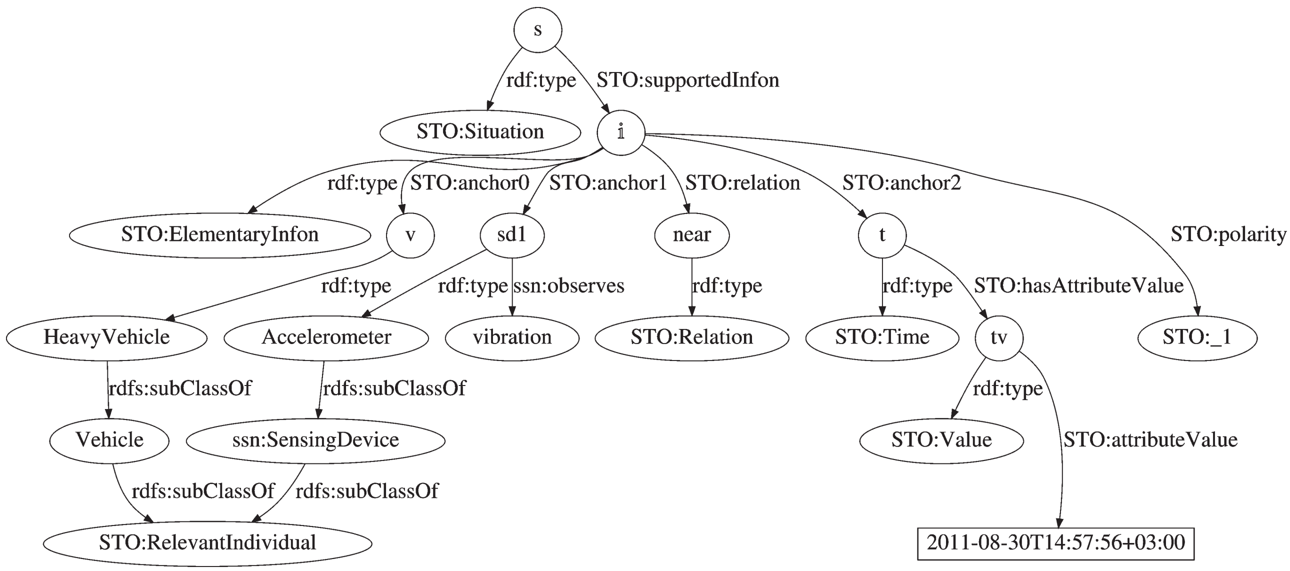

Fig. 2. RDF graph describing the situation `s` at time 14:57:56+03:00 on August 30, 2011, whereby the heavy vehicle `v` was `near` to the accelerometer sensing device `sd1`.

IV. Discussion

For the case of vehicle detection and classification by measurement of road pavement vibration, we have shown how the classification of vibration patterns induced by entities observable by a sensor network can be translated into the representation of knowledge for situations involving the entities.

We represented knowledge for the situation whereby a vehicle is *near* an accelerometer sensing device; for the relevant individuals in such situations, namely light vehicles, heavy vehicles, and accelerometer sensing devices, which observe *vibration*, a property of the feature *pavement*; for the disjointness of light and heavy vehicles; and for the relations *driving-side* and *driving-speed*. Further, we implemented two domain rules to infer: 1) when two vehicles are the same physical object; and 2) a vehicle's driving side and speed. We described the infrastructure used to retrieve road pavement vibration data, how such data are processed to detect possible vehicle occurrences and to construct labeled data sets, how the data sets are used to train artificial neural network classifiers, and how we represented observations and situations, consistent with the SSNO and the STO, respectively.

As defined by the ontology (see Fig. 3), our domain consists of light and heavy vehicles. Therefore, each vehicle type identified in camera data was mapped to the concept of light vehicle or the concept of heavy vehicle. For the training and evaluation of MLP classifiers, we used the mapping from the actual vehicle label to the training class shown in Table II. For the system evaluation between 2:50 P.M. and 3:00 P.M., all vehicle individuals involved in the 12 situations were necessarily instances of either light or heavy vehicle. Entirely new vehicle types, e.g., motorcycles, would be also classified either as light or heavy vehicle. Thus, it is the ontology that defines what is known to the system.

In addition to vibration data, the system could also use camera data for the purpose of vehicle detection and classification. Knowledge could be acquired and represented from camera data, and we may then relate it to knowledge acquired from vibration data. Alternatively, camera data may be used together with vibration data in a single machine learning step. In our setup, camera data were not meant to be used in machine learning. Instead, the aim was to classify vehicles based on vibration data only.

Listing 1. Axioms with domain knowledge

Accelerometer $\sqsubseteq$ ssn:SensingDevice
ssn:SensingDevice $\sqsubseteq$ sto:RelevantIndividual
Vehicle $\sqsubseteq$ sto:RelevantIndividual
LightVehicle $\sqsubseteq$ Vehicle
HeavyVehicle $\sqsubseteq$ Vehicle
LightVehicle $\sqcap$ HeavyVehicle $\sqsubseteq \bot$

sto:Relation($near$)
sto:Relation(*driving-side*)
sto:Relation(*driving-speed*)
ssn:FeatureOfInterest($pavement$)
ssn:Property($vibration$)
ssn:isPropertyOf($vibration, pavement$)
Accelerometer(sd_1)
Accelerometer(sd_2)
Accelerometer(sd_3)
ssn:observes($sd_1, vibration$)
ssn:observes($sd_2, vibration$)
ssn:observes($sd_3, vibration$)

Fig. 3. Listing 1. Axioms with domain knowledge.

The presented system was evaluated over 10 min of *historical* data. Hence, while the use case may be developed further to include a production system that performs near real-time situational knowledge representation, we have only simulated a real-time context. At the end of the project related to this paper, we were unable to access the sensor network. As we performed this work after the project ended, we could not evaluate the presented system in a true real-time context. Moreover, the samples at our disposal for testing and evaluation purposes were limited.

The system architecture shown in Fig. 1 supports specific implementations for measurement engines, observation engines and stores, situation engines and stores, knowledge stores, as well as digital signal processing, machine learning, complex event processing, and inference modules. Measurement engines and situation engines, including associated modules, are domain specific. Indeed, the software logic necessary to acquire and decode data from sensing devices is device specific. Moreover, measurement engines can include files, databases, or distributed resources, in addition to sensors. Similarly, situations acquired from observations are domain specific. Hence, the implementation of situation engines is domain specific. In the present form, the implementation of situational knowledge acquisition tasks may be thus somewhat tedious. In future work, we aim at proposing a solution whereby such knowledge acquisition tasks can be driven via ontology. In contrast, the implementation of an observation engine can be more generic. However, specific use cases may want to make use of (rule-based) inference in order to further semantically enrich observations. Knowledge stores are specific to OWL knowledge bases. The system currently supports persisting observations and situations using knowledge stores backed by the OWL API[10] and the Stardog RDF database.[11] The system can be extended with knowledge store implementations for other OWL knowledge bases.

We implemented rule inference of same vehicles and of vehicle velocity manually, in Java, as processes performed for situations occurring between 2:50 P.M. and 3:00 P.M. Hence, we first acquired knowledge about situations, and then, we performed inference. We also tested the implementation and evaluation of the rules using the SWRL, but ultimately, we did not succeed due to technical limitations with respect to SWRL expressivity or language implementation by reasoners. For instance, we implemented our first rule for same vehicles in Protégé using the SWRL built-in `subtractDateTimesYieldingDayTimeDuration`. However, the SWRL built-in was not supported by reasoners in Protégé (Hermit, FaCT++, or Pellet). Naturally, a declarative implementation of rules, along with their evaluation by reasoners, is desirable. The applicability of declarative languages for the inference of situational knowledge, in particular using the STO, as well as the scalability of their evaluation using reasoners, is largely an open question. We plan to address some of these issues in future work.

Representing symbolic knowledge acquired from sensor data for situations that involve entities observable by a sensor network has a number of interesting implications. First, similarly to other systems that build on sensors and acquire knowledge, we abstract from sensor data. As we have shown, our small sensor network consisting of three sensors, each sampling at 2 kHz, generated approximately 130 million values in a mere six hours of measurement. Such data can be a challenge to manage and is of little interest if not for the knowledge conveyed by them, particularly knowledge for situations involving vehicles. We argue that the persistence of situational knowledge can be a desirable alternative to the persistence of sensor data, which can be thus discarded.

Second, we explicitly and formally represent knowledge acquired from sensor networks in ontology. The result is a unified representation of (inferred) knowledge about situations monitored by a sensor network. Hence, ontology facilitates the integration of knowledge acquired in sensor networks and the automatic discovery of new knowledge. Several arguments speak for integration and inference at the situation layer. Clearly, vehicle velocity can be also computed using digital signal processing on data of two (or more) sensors. Given that our system has access to data of all three sensors, this is indeed an alternative. However, we may envision a use case whereby a system only processes the data of one sensor. The resulting three systems would not know about each others' sensor data, let alone have access to it. Each system would generate situational knowledge, individually. Represented knowledge may be thus integrated to provide a view for the situations acquired by the three systems. We can thus perform inference over such an integrated view. (Note that this description is not unlike how we implemented in fact the described use case.) More generally speaking, systems may be distributed, of varying complexity and proprietary. In such cases, it may be challenging to integrate at the measurement layer. Moreover, integration and inference at the situation layer may be more straightforward than at the measurement layer because users are closer to the domain (i.e., it becomes possible to work with domain concepts). However, the situation (more generally speaking, ontology) layer is not a silver bullet: While some problems may be more elegantly solved at the situation layer, others may not be solvable at the situation layer.

Third, both users and computers may interact with knowledge-rich systems rather than data-rich systems. For instance, we may now use an RDF query language and system to query, for example, situations involving vehicles (both light and heavy) that drove on the right roadside at speed between 20 and 40 km/h on August 30, 2011, between 3:00 P.M. and 4:00 P.M. By using ontology to represent knowledge about situations monitored by a sensor network, a system commits to reusing domain terminology explicitly defined outside of the system. Indeed, most of the terminology used here is defined in the SSNO and the STO. We only trivially extended these ontologies to accommodate domain knowledge. The commitment to reusing domain terminology explicitly represented in ontology can increase the interoperability of systems [27], [28]. We argue that this is a key benefit of ontology-based systems.

Fourth, we highlight a possible role of ontology consistency checking. As shown in Table V, there are three vehicle occurrences (14:50, 14:53, and 14:58) for which machine learning is in disagreement regarding the specific class of the vehicle involved in the respective situations. This suggests that, in those three occurrences, the three situations, respectively, involve different vehicles. However, based on the temporal location at which vehicles are near sensors in different situations, rule inference suggests that, in those occurrences, the respective three situations involve the same vehicles, e.g., at 14:50 a personal car. Hence, inference could conclude that these vehicles are both light and heavy. This, however, contradicts the disjointness

[10]http://owlapi.sourceforge.net/

[11]http://www.stardog.com/

axiom that states that a vehicle cannot be light and heavy. An OWL reasoner would hence determine that the ontology is inconsistent. It could also explain which OWL axioms are inconsistent. In our case, it can be shown that an ontology including only the vehicle occurrence at 14:57 is consistent. The question arises how to resolve such inconsistencies. Generally speaking, resolution may require user feedback or may be carried out automatically. In our case, if machine learning fails, meaning that, for the same physical entity, the vehicle classes of three situations are in disagreement, we could resolve the conflict by choosing the vehicle class with most votes by classifiers. Explanation services could identify the inconsistent ontology axioms and, hence, the vehicle individuals of situations that are in conflict. Using majority voting, the system could resolve such conflicts automatically. On the other hand, if the `sameAs` rule fails, meaning that, in reality, for three temporally close situations, (for instance) two involve the (physically) same light vehicle, and the third involves a (physically) different heavy vehicle, then resolution may require manual assessment, e.g., a correction in rule parametrization. More sophisticated automated conflict resolution techniques may be of interest to solving such inconsistencies.

Fifth, we underscore a few arguments that motivate our choice for an ontology-based approach, in contrast to classical relational database systems. First, with the ontology approach, we can focus on the modeling of domain knowledge and semantics, and leave the data modeling to the knowledge base. Second, we build our system on two readily available ontologies. Such terminologies can support the modeling of domain knowledge since they provide an organization of relevant generic concepts and relations. Moreover, they can guide the design and implementation of software systems. In fact, we greatly aligned our system implementation to relevant concepts and relations of both the SSNO and the STO. Hence, the system is generic, and its implementation can be reused across domains. Finally, in future work, we also aim at showing that scalable solutions for the discussed problem can be designed and implemented using technologies other than classical relational databases.

V. Related Work

Classifying objects or events observed by sensors has a long tradition and extensive literature, diversified according to observation and methodology. For a multitude of sensors, an array of techniques, including vision-based techniques, acoustic signature analysis, vehicle axle counting, and pneumatic road tubes, have been proposed alone for the detection and classification of road vehicles [29]–[32].

The relevance of ontologies to sensor networks is evident from the literature. Sheth *et al.* [33] present the Semantic Sensor Web in which sensor data are annotated with spatial, temporal, and thematic semantic metadata. Terminologies [6], [34] to describe the characteristics of sensors and sensor networks have received considerable attention. Compton *et al.* [5] review 11 sensor ontologies for their range and expressive power.

The semantic enrichment of sensor data has received considerable attention. Wanner *et al.* [35] present an environmental information system in which environmental data, e.g., temperature measurements for a city, are stored in an OWL knowledge base [36]. Barnaghi *et al.* [37] propose a semantic model for (heterogeneous) sensor data representation, discussed using both Extensive Markup Language (XML) and OWL. Wei and Barnaghi [38] annotate sensor data with semantic metadata and relate sensor data with data of knowledge bases available online, such as DBpedia [39], following the linked data principle [40].

An advantage of semantic enrichment of sensor data is that, in doing so, systems can leverage ontology and rule-based reasoning to infer new knowledge from sensor data. This practice is well documented, for instance by Sheth *et al.* [33], to infer a "blizzard condition"; by Henson *et al.* [41], to infer "high winds" observations; by Stocker *et al.* [42], to infer the nutrient status of lakes; and by Wei and Barnaghi [38], to infer the approximate temperature for a city neighboring a city for which the temperature is known. If a knowledge acquisition task on sensor data is within the expressivity of the languages used and can be hence formally represented by means of said languages, representing sensor data in ontology is attractive as it allows us to leverage the powerful reasoning capabilities of inference engines. However, not all knowledge acquisition tasks on sensor data are within the expressivity of state-of-the-art ontology and rule languages. For instance, the knowledge acquisition task discussed here for the classification of vehicles using road pavement vibration data and machine learning cannot be formalized in an OWL ontology (the language lacks of a notion for Fourier transform, for example).

Using the SSNO, Barnaghi *et al.* [8] describe a framework aimed at creating perception from sensor data. They motivate their work by underscoring how data consumers "are often interested in the higher level concepts, such as events," rather than low-level sensor data. Contrary to [8] and our previous work [43], [44], here, we make use of the STO to represent knowledge at the most abstract level, leaving the SSNO at an intermediate level for the semantic enrichment of sensor data.

This paper relates to the work of Fenza *et al.* [45], who use the STO to represent airport security situations; of De Maio *et al.* [46], who present an approach to identify situations represented using the STO; and of Doulaverakis *et al.* [47], who use the STO in an architecture for intelligent information fusion in a sensor network environment, which the authors demonstrate for the domain of security and surveillance. In contrast to this related work, we suggest adopting the SSNO and the STO at different levels of abstraction. Inference on, as well as storage of, observations motivates the intermediate layer of semantic enrichment of measurements.

Conroy *et al.* [12], [48] extract from sensor data various biological and physiological properties of athletes during exercise. Contrary to their approach based on XML, we suggest using an expressive ontology language to formalize acquired knowledge. Gaglio *et al.* [49] propose a generic architecture to extract information from an environment sensed by a wireless sensor network and discuss it for a case study on wildfire detection. The generic architecture presented by Gaglio *et al.* relates to the three-layered architecture presented here, in that both aim at bridging the measurement layer with the knowledge layer. In contrast to Gaglio *et al.*, we ground the symbolic layer in situation theory [18]. Liu and Zhao [7] and Whitehouse *et al.* [50] discuss the architecture of a system that can be queried for high-level events without requiring handling of raw magnetometer data, specifically for a parking garage case study. The authors elaborate on a programming model called Semantic Streams, which rests on two fundamental elements, with *event streams* being one of them. Hence, at the base of Semantic Streams are detected entities such as objects, people, or events. In contrast, we do not assume such a stream as given.

VI. Conclusion

For the case of vehicle detection and classification by measurement of road pavement vibration, we have shown, using digital signal processing and machine learning, how knowledge about situations observable by a sensor network can be acquired from sensor data and can be formally represented by means of ontology. We have presented and discussed the process from sensor data acquisition to knowledge representation.

Specifically, we used digital signal processing to process sensor data and machine learning to acquire knowledge about the physical entities involved in situations monitored by a sensor network. We formally represented such knowledge in a domain ontology that borrows from both the SSNO and the STO. Rules were formulated to infer new knowledge about situations, e.g., vehicle velocity. We have discussed a number of implications that result from representing knowledge acquired from sensor network data in ontology, particularly abstraction from sensor data, integrated representation of knowledge about monitored situations, rule inference, querying in knowledge-rich systems, and ontology consistency checking and explanation.

We have discussed a three-layered software architecture for a system aimed at continuous, distributed, and (near) real-time processing of sensor network data as well as at the acquisition and representation of situational knowledge. The system abstracts from sensor data by semantic enrichment of measurements to observations and by acquisition and representation of situations from observations. Hence, the system reduces the gap between sensor data and abstract terminology used by people to describe real-world situations. The SSNO is used as upper ontology for the representation of domain-specific observations, as well as knowledge related to devices and sensing. The STO is used as upper ontology for the representation of domain-specific situations. Being grounded in situation theory, the STO seems to provide a useful terminology for the representation of knowledge about real-world situations, including for intelligent transportation systems. The hybrid application of both the SSNO and the STO enables the adoption of appropriate terminology at each level of abstraction, as well as flexible storage of observations or situations according to application requirements.

The discussed case highlights that some knowledge acquisition tasks on sensor data are beyond the expressivity of state-of-the-art ontology and declarative rule languages, as well as reasoners. Hence, such tasks cannot be formalized by means of declarative rules. Instead, the acquired knowledge is a result of computations in digital signal processing, machine learning, or complex event processing.

Acknowledgment

We wish to thank Paula Silvonen, research scientist at VTT Technical Research Center of Finland, for her valuable contribution to related work. This work was conducted using Protégé (grant LM007885, U.S. National Library of Medicine).

References

[1] L. Finkelstein, "Theory and philosophy of measurement," in *Handbook of Measurement Science*, P. H. Sydenham, Ed. Hoboken, NJ, USA: Wiley, 1982, pp. 1–30.

[2] P. Bonnet, J. Gehrke, and P. Seshadri, "Towards sensor database systems," in *Mobile Data Management*. Berlin, Germany: Springer-Verlag, 2001, pp. 3–14.

[3] D. Carney, U. Çetintemel, M. Cherniack, C. Convey, S. Lee, G. Seidman, M. Stonebraker, N. Tatbul, and S. Zdonik, "Monitoring streams: a new class of data management applications," in *Proc. 28th Int. Conf. VLDB*, 2002, pp. 215–226, VLDB Endowment.

[4] S. Madden and M. Franklin, "Fjording the stream: an architecture for queries over streaming sensor data," in *Proc. 18th Int. Conf. Data Eng.*, 2002, pp. 555–566.

[5] M. Compton, C. Henson, H. Neuhaus, L. Lefort, and A. Sheth, "A survey of the semantic specification of sensors," presented at the 2nd Int. Workshop Semantic Sensor Networks, 8th Int. Semantic Web Conf., 2009.

[6] M. Eid, R. Liscano, and A. El Saddik, "A novel ontology for sensor networks data," in *Proc. IEEE Int. Conf. Comput. Intell. Meas. Syst. Appl.*, Jul. 2006, pp. 75–79.

[7] J. Liu and F. Zhao, "Towards semantic services for sensor-rich information systems," in *Proc. 2nd Int. Conf. BroadNets*, Oct. 2005, vol. 2, pp. 967–974.

[8] P. Barnaghi, F. Ganz, C. Henson, and A. Sheth, Computing perception from sensor data, Kno.e.sis, Dayton, OH, USA, Tech. Rep., 2012. [Online]. Available: knoesis.org

[9] D. J. Hill, Y. Liu, L. Marini, R. Kooper, A. Rodriguez, J. Futrelle, B. S. Minsker, J. Myers, and T. McLaren, "A virtual sensor system for user-generated, real-time environmental data products," *Environ. Modell. Softw.*, vol. 26, no. 12, pp. 1710–1724, Dec. 2011.

[10] M. Balazinska, A. Deshpande, M. Franklin, P. Gibbons, J. Gray, S. Nath, M. Hansen, M. Liebhold, A. Szalay, and V. Tao, "Data management in the worldwide sensor web," *IEEE Pervasive Comput.*, vol. 6, no. 2, pp. 30–40, Apr.–Jun. 2007.

[11] D. Kirsh, "When is information explicitly represented," *Vancouver Studies Cogn. Sci.*, pp. 340–365, 1990.

[12] K. Conroy, G. May, M. Roantree, G. Warrington, S. J. Cullen, and A. McGoldrick, "Knowledge acquisition from sensor data in an equine environment," in *Proc. 13th Int. Conf. DaWaK*, 2011, pp. 432–444.

[13] S. Haykin, *Neural Networks: A Comprehensive Foundation*. Upper Saddle River, NJ, USA: Prentice-Hall, 1999, ser. Prentice Hall International Editions Series.

[14] K. Janowicz and M. Compton, "The stimulus-sensor-observation ontology design pattern and its integration into the semantic sensor network ontology," in *Proc. 3rd Int. Workshop Semantic Sensor Netw.*, 2010, pp. 7–11.

[15] M. Compton, "What now and where next for the W3C semantic sensor networks incubator group sensor ontology," in *Proc. 4th Int. Workshop SSN11*, 2011, pp. 1–8.

[16] M. M. Kokar, C. J. Matheus, and K. Baclawski, "Ontology-based situation awareness," *Inf. Fusion*, vol. 10, no. 1, pp. 83–98, Jan. 2009.

[17] J. Barwise and J. Perry, *Situations and Attitudes*. Cambridge, MA, USA: MIT Press, 1983, ser. Bradford Books.

[18] K. Devlin, *Logic and Information*. Cambridge, U.K.: Cambridge Univ. Press, 1995.

[19] M. Endsley, "Toward a theory of situation awareness in dynamic systems," *Hum. Factors, J. Human Factors Ergonom. Soc.*, vol. 37, no. 1, pp. 32–64, Mar. 1995.

[20] M. Endsley, "Theoretical underpinnings of situation awareness: A critical review," in *Situation Awareness Analysis and Measurement*. Mahwah, NJ, USA: Lawrence Erlbaum Associates, 2000, pp. 3–32.

[21] *AXIS 211W Network Camera Datasheet*, Axis Communications, Lund, Sweden, 2011.

[22] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: An update," *SIGKDD Explorations*, vol. 11, no. 1, pp. 10–18, Jun. 2009.

[23] "OWL 2 Web ontology language," W3C, Cambridge, MA, USA, Tech. Rep. W3C Recommendation, Oct. 27, 2009.

[24] F. Manola, E. Miller, and B. McBride, "RDF primer," W3C, Cambridge, MA, USA, W3C Recommendation, Feb. 10, 2004.

[25] J. J. Carroll, I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne, and K. Wilkinson, "Jena: Implementing the semantic web recommendations," HP Lab., Bristol, U.K., Tech. Rep. HPL-2003-146, Dec. 24, 2003.

[26] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean, SWRL: A Semantic web rule language combining OWL and RuleML, W3C, Cambridge, MA, USA, Tech. Rep., May 21, 2004. [Online]. Available: http://www.w3.org/Submission/SWRL/

[27] L. Obrst, "Ontologies for semantically interoperable systems," in *Proc. 12th Int. CIKM*, 2003, pp. 366–369.

[28] M. Uschold and M. Gruninger, "Ontologies and semantics for seamless connectivity," *SIGMOD Rec.*, vol. 33, no. 4, pp. 58–64, Dec. 2004.

[29] R. Bajwa, R. Rajagopal, P. Varaiya, and R. Kavaler, "In-pavement wireless sensor network for vehicle classification," in *Proc. 10th Int. Conf. IPSN*, 2011, pp. 85–96.

[30] J. Jackowski and R. Wantoch-Rekowski, "Classification of wheeled military vehicles using neural networks," in *Proc. 18th ICSEng*, 2005, pp. 212–217.

[31] W. Wu, Z. QiSen, and W. Mingjun, "A method of vehicle classification using models and neural networks," in *Proc. IEEE VTS 53rd VTC-Spring*, 2001, vol. 4, pp. 3022–3026.

[32] A. Y. Nooralahiyan, M. Dougherty, D. McKeown, and H. R. Kirby, "A field trial of acoustic signature analysis for vehicle classification," *Transp. Res. C, Emerg. Technol.*, vol. 5, no. 3/4, pp. 165–177, Aug./Oct. 1997.

[33] A. Sheth, C. Henson, and S. Sahoo, "Semantic Sensor Web," *IEEE Internet Comput.*, vol. 12, no. 4, pp. 78–83, Jul./Aug. 2008.

[34] S. Avancha, C. Patel, and A. Joshi, "Ontology-driven adaptive sensor networks," in *Proc. 1st Annu. Int. Conf. MOBIQUITOUS*, Aug. 2004, pp. 194–202.

[35] L. Wanner, S. Vrochidis, S. Tonelli, J. Mograber, H. Bosch, A. Karppinen, M. Myllynen, M. Rospocher, N. Bouayad-Agha, U. Bügel, G. Casamayor, T. Ertl, I. Kompatsiaris, T. Koskentalo, S. Mille, A. Moumtzidou, E. Pianta, H. Saggion, L. Serafini, and V. Tarvainen, "Building an environmental information system for personalized content delivery," in *Environmental Software Systems. Frameworks of eEnvironment*, J. Hrebícek, G. Schimak, and R. Denzer, Eds. Boston, MA, USA: Springer-Verlag, 2011, ser. IFIP Advances in Information and Communication Technology, pp. 169–176.

[36] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, *The Description Logic Handbook: Theory, Implementation and Applications.*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2007.

[37] P. Barnaghi, S. Meissner, M. Presser, and K. Moessner, "Sense and sens'ability: Semantic data modelling for sensor networks," in *Proc. ICT Mobile Summit*, 2009, vol. 2009, pp. 1–9.

[38] W. Wei and P. Barnaghi, "Semantic annotation and reasoning for sensor data," in *Proc. Smart Sens. Context*, 2009, pp. 66–76.

[39] C. Bizer, S. Auer, G. Kobilarov, J. Lehmann, and R. Cyganiak, "DBpedia–Querying Wikipedia like a database," in *Proc. 16th Int. Conf. WWW*, 2007, pp. 8–12, Developers track presentation.

[40] T. Berners-Lee, Linked Data–Design Issues, 2006. [Online]. Available: http://www.w3.org/DesignIssues/LinkedData.html

[41] C. A. Henson, J. K. Pschorr, A. P. Sheth, and K. Thirunarayan, "SemSOS: Semantic sensor observation service," in *Proc. 2009 Int. Symp. CTS*, Baltimore, MD, USA, May 2009, pp. 44–53.

[42] M. Stocker, M. Rönkkö, F. Villa, and M. Kolehmainen, "The relevance of measurement data in environmental ontology learning," in *Environmental Software Systems. Frameworks of eEnvironment*. Boston, MA, USA: Springer-Verlag, 2011, ser. IFIP Advances in Information and Communication Technology, pp. 445–453.

[43] M. Stocker, M. Rönkkö, and M. Kolehmainen, "Making sense of sensor data using ontology: A discussion for residential building monitoring," in *Artificial Intelligence Applications and Innovations*, L. Iliadis, I. Maglogiannis, H. Papadopoulos, K. Karatzas, and S. Sioutas, Eds. Boston, MA, USA: Springer-Verlag, 2012, ser. IFIP Advances in Information and Communication Technology, pp. 341–350.

[44] M. Stocker, M. Rönkkö, and M. Kolehmainen, "Making sense of sensor data using ontology: A discussion for road vehicle classification," in *Proc. Int. Congr. Environ. Model. Softw.*, R. Seppelt, A. Voinov, S. Lange, and D. Bankamp, Eds., Leipzig, Germany, Jul. 2012, pp. 2387–2394.

[45] G. Fenza, D. Furno, V. Loia, and M. Veniero, "Agent-based cognitive approach to airport security situation awareness," in *Proc. Int. Conf. CISIS*, 2010, pp. 1057–1062.

[46] C. De Maio, G. Fenza, D. Furno, and V. Loia, "Swarm-based semantic fuzzy reasoning for situation awareness computing," in *Proc. Int. Conf. FUZZ-IEEE*, Jun. 2012, pp. 1–7.

[47] C. Doulaverakis, N. Konstantinou, T. Knape, I. Kompatsiaris, and J. Soldatos, "An approach to intelligent information fusion in sensor saturated urban environments," in *Proc. EISIC*, Sep. 2011, pp. 108–115.

[48] K. Conroy, G. C. May, M. Roantree, and G. Warrington, "Expanding sensor networks to automate knowledge acquisition," in *Proc. 28th BNCOD*, 2011, pp. 97–107.

[49] S. Gaglio, L. Gatani, G. Lo Re, and M. Ortolani, "Understanding the environment through wireless sensor networks," in *Proc. 10th Congr. AI*IA, Artif. Intell. Hum.Orient. Comput.*, 2007, pp. 72–83.

[50] K. Whitehouse, F. Zhao, and J. Liu, "Semantic streams: A framework for composable semantic interpretation of sensor data," in *Wireless Sensor Networks*, K. Römer, H. Karl, and F. Mattern, Eds. Berlin, Germany: Springer-Verlag, 2006, ser. LNCS, pp. 5–20.

Markus Stocker was born in Switzerland in 1979. He received the M.Sc. degree in informatics from University of Zurich, Zurich, Switzerland, in 2007. He is currently working toward the M.Sc. degree in environmental science and the Ph.D. degree in environmental technology at the Department of Environmental Science, University of Eastern Finland, Kuopio, Finland.

In 2007, he was a Research Intern with HP Laboratories, Bristol, U.K., where he designed and developed a query optimizer. In 2008, he was a Research Intern with Clark & Parsia LLC, Washington, D.C., USA, where he was involved in a number of projects, with focus on semantic web technologies. Since 2009, he has been a Research Associate with University of Eastern Finland and a member of the Environmental Informatics Research Group. His research interests include knowledge acquisition and representation from sensor data.

Mauno Rönkkö was born in Finland in 1970. He received the M.Sc. degree in computer science from University of Kuopio (currently University of Eastern Finland), Kuopio, Finland, in 1995 and the Ph.D. degree in computer science from Åbo Akademi University, Turku, Finland, in 2001.

Since 2010, he has been a Project Manager with the Environmental Informatics Research Group, University of Eastern Finland. His research interests include environmental monitoring, particularly environmental safety and security applications.

Mikko Kolehmainen was born in Finland in 1962. He received the M.Sc. degree in software engineering and the D.Sc. degree in computational methods from Helsinki University of Technology, Helsinki, Finland, in 1987 and 2004, respectively, and the M.Sc. degree in biotechnology from University of Kuopio (currently University of Eastern Finland), Kuopio, Finland, in 2001.

He started his career with a private software company as an Analyst and Development Manager. Since 1996, he has been with University of Eastern Finland, first as a Researcher, then as a Research Director since 2002, and as a Professor of environmental informatics since 2006. Under his guidance, the research group has successfully carried out both national and European Union projects. His main research and teaching interests include the application of neural networks to environmental data and methods of bioinformatics in environmental monitoring.

Dr. Kolehmainen served as the Chair for the Ninth International Conference on Adaptive and Natural Computing Algorithms in 2009 and continues to work on its Advisory Committee.