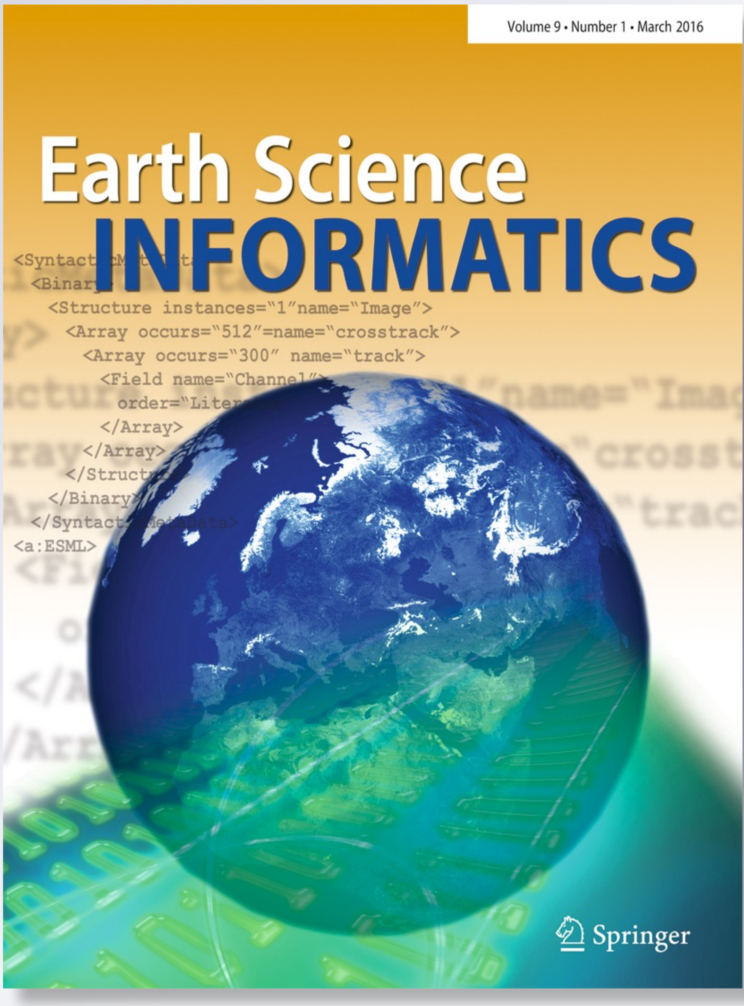
Knowledge-based environmental research infrastructure: moving beyond data

Markus Stocker, Mauno Rönkkö & Mikko Kolehmainen

Earth Science Informatics

ISSN 1865-0473
Volume 9
Number 1

Earth Sci Inform (2016) 9:47-65
DOI 10.1007/s12145-015-0230-6



Springer

Earth Sci Inform (2016) 9:47–65
DOI 10.1007/s12145-015-0230-6

RESEARCH ARTICLE

Knowledge-based environmental research infrastructure: moving beyond data

Markus Stocker[1] · Mauno Rönkkö[1] · Mikko Kolehmainen[1]

Received: 18 January 2015 / Accepted: 25 May 2015 / Published online: 30 June 2015

Abstract Over the past decades, sensor networks have been deployed around the world to monitor over time and space a large number of properties appertaining to various environmental phenomena. A popular example is the monitoring of particulate matter and gases in ambient air undertaken, for instance, to assess air quality and inform decision makers and the public. Such infrastructure can generate large amounts of data, which must be processed to derive useful information. Infrastructure may be for environmental research, specifically. In order to reduce duplication and improve interoperability, efforts have been initiated more recently that aim at abstract architectural descriptions of infrastructure that supports the acquisition, curation, access, and processing of measurement and observation data. The ENVRI Reference Model (ENVRI-RM) is an example for an abstract architectural description of infrastructure tailored for environmental research. We briefly summarize ENVRI-RM and provide an overview of its subsystems, functionality, and viewpoints. We highlight that its primary focus is on the data life-cycle in environmental research infrastructure. As our contribution, we extend ENVRI-RM with functionality for the acquisition of knowledge from data, and the curation, access, and processing of knowledge. The extension, which we name +K, aims at addressing the knowledge life-cycle in environmental research infrastructure. We present the +K subsystems and functionality, and discuss the extension from ENVRI-RM viewpoints. We argue that the +K extension can be superimposed on ENVRI-RM to form the ENVRI-RM+K model for the 'archetypical' knowledge-based environmental research infrastructure that addresses both data and knowledge life-cycles. We demonstrate the application of the extension to a concrete use case in aerosol science.

Communicated by: H. A. Babaie

✉ Markus Stocker
markus.stocker@uef.fi

Mauno Rönkkö
mauno.ronkko@uef.fi

Mikko Kolehmainen
mikko.kolehmainen@uef.fi

[1] Research Group of Environmental Informatics, Department of Environmental Science, University of Eastern Finland, P.O. Box 1627, 70211, Kuopio, Finland

Keywords Environmental research infrastructure · Knowledge-based systems · Knowledge acquisition · Knowledge representation and reasoning · Ontology · Semantic web technologies

Introduction

Environmental research infrastructure, also known as environmental cyberinfrastructure, are complex and distributed hardware and software systems that collect environmental monitoring data and manage such data (primarily) for research. The development of environmental research infrastructure is on the agenda of EU and national funding bodies, in particular also in the US. Transatlantic collaborations, such as COOPEUS,[1] further underscore the interest in environmental research infrastructure.

Environmental research infrastructure often builds on environmental sensor networks (Martinez et al. 2004; Hart and Martinez 2006) which is hardware infrastructure that

[1] http://www.coopeus.eu/

produces data. Software infrastructure then provides services that support the acquisition of data produced by environmental sensor networks, and the curation, access, and processing of data. The National Ecological Observatory Network (Keller et al. 2008), the Global Lake Ecological Observatory Network (Kratz et al. 2006), and the Polar Cyberinfrastructure (Li et al. 2014) are just a few recent examples for diverse environmental research infrastructure that have been designed and (are being) deployed.

In Finland, the Station for Measuring Ecosystem-Atmosphere Relations[2] (SMEAR) is an example for an environmental sensor network that primarily serves research (Hari and Kulmala 2005). SMEAR started its operations in 1991 with measurements for SO_2 in Eastern Lapland. The station quickly grew to include several other locations and environmental phenomena, including weather, aerosols, photosynthesis, and solar radiation. Today, data collected by SMEAR are used towards various research purposes, including the study of atmospheric new particle formation (Dal Maso et al. 2005). Software systems have been developed to manage and visualize SMEAR data (Junninen et al. 2009). Another national example is Soil-Weather, a wireless sensor network that aims at providing "temporally and spatially accurate information, data services and (real-time) applications for water monitoring and agriculture on river basin and farm scales" (Kotamäki et al. 2009).

Such hardware and software systems have historically been constructed *ad hoc*, designed for a specific problem. This has lead to great heterogeneity among them, which is now hindering their integration, in particular the integration of data. Furthermore, individual systems have addressed common technical challenges independently, resulting in duplication of effort.

The EU FP7 Common Operations of Environmental Research Infrastructures (ENVRI) project[3] aimed at addressing these concerns by developing data and software components and services common to six European Strategy Forum on Research Infrastructures (EU ESFRI) environmental research infrastructures. These are the Integrated Carbon Observation System (ICOS); the European contribution to the Argo program for global ocean monitoring (EURO-Argo); the European Incoherent Scatter Scientific Association for the study of the atmosphere in the Fenno-Scandinavian Arctic and its coupling to space (EISCAT-3D); the E-Science European Infrastructure for Biodiversity and Ecosystem Research (LifeWatch); the European Plate Observing System (EPOS); and the European Multidisciplinary Seafloor & Water Column Observatory (EMSO). ENVRI aimed at identifying computational characteristics common to environmental research infrastructure, develop an understanding of requirements, support and accelerate the construction of infrastructure, secure interoperability between infrastructures, avoid duplication of effort, and enable the reuse of resources and experiences (Chen et al. 2013a, b).

The ENVRI *Reference Model* (ENVRI 2013), hereafter ENVRI-RM, is arguably the primary result of the ENVRI project. ENVRI-RM is "a common ontological framework and standard for the description and characterization of computational and storage infrastructures" and provides "a universal reference framework for discussing many common technical challenges facing all of the ESFRI environmental research infrastructures" (Chen et al. 2013b). ENVRI-RM was a collaborative effort between several institutions, including Cardiff University, University of Edinburgh, European Environment Agency, and University of Amsterdam. The reference model is publicly available[4] and its latest version is V1.1 of August 30, 2013.

ENVRI-RM addresses challenges that are common to state-of-the-art environmental research infrastructure, in particular challenges related to the acquisition of (streamed) data; the management and archiving of acquired data in databases and on storage infrastructure; the discovery and access of data from distributed data archives; the integration, harmonization, and publication of data; the processing of data in scientific workflows and for visualization, statistical analysis, and data mining; and the construction of linked, distributed, software and hardware infrastructure. Chen et al. (2013b) note that ENVRI-RM is the first model of its kind, and the authors underscore the urgency of a reference model for environmental research infrastructure, in particular because it is "[o]nly by adopting a good reference model [that the community can] secure interoperability between infrastructures, enable reuse, share resources and experiences, and avoid unnecessary duplication of effort."

ENVRI-RM makes evident that the six EU ESFRI environmental research infrastructures are primarily concerned with data, and services for data acquisition, curation, access, and processing. This conclusion is arguably more widely applicable to environmental research infrastructure other than the six analyzed within ENVRI.

As our contribution, we propose to extend ENVRI-RM with functionality for the acquisition of knowledge from data, and the curation, access, and processing of knowledge. Our claim is that, beyond data, environmental research infrastructure can, and perhaps should, manage

[2] http://www.atm.helsinki.fi/SMEAR/

[3] http://envri.eu/

[4] http://www.envri.eu/rm

knowledge derived from processed data curated within the infrastructure. The result is knowledge-based environmental research infrastructure.

We begin with a short overview of ENVRI-RM. We then present the proposed ENVRI-RM extension, and discuss some technical aspects of the extension. Finally, we demonstrate the application of the extension to a concrete use case in aerosol science.

Reference model

This section provides readers with a summary of the ENVRI Reference Model (ENVRI 2013). The purpose of the summary is to build the context for the subsequently presented *extension* to ENVRI-RM, and to highlight that data is the primary concern of ENVRI-RM. We first discuss how ENVRI-RM divides environmental research infrastructure into subsystems and partitions common functionality amongst the subsystems. Then we discuss the three viewpoints—science, information, and computational—from which ENVRI-RM is defined.

Reference model subsystems

ENVRI-RM divides the 'archetypical' environmental research infrastructure into five subsystems: data acquisition, data curation, data access, data processing, and community support. Figure 1 represents the ENVRI-RM subsystems graphically. The subdivision broadly follows the data life-cycle in environmental research infrastructure. According to ENVRI (2013), the data life-cycle "begins with the acquisition of raw data from a network of integrated data collecting instruments." Instruments are understood broadly to include both devices and human observers. Acquired raw data is then pre-processed and curated within the infrastructure. Curated data is made accessible, to humans

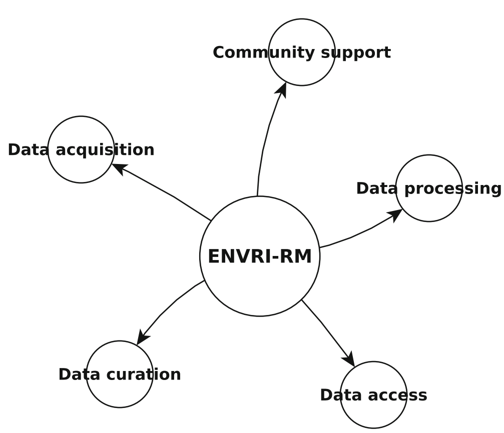


Fig. 1 The ENVRI-RM subsystems

and services. Data access enables data to "be extracted from parts of the infrastructure and made subject to data processing." Community support is orthogonal to, and cuts across, the other four subsystems (Chen et al. 2013a).

ENVRI-RM identifies common functionality, which is partitioned amongst the five subsystems. The reference model includes a graphical representation of functionality partitioning (ENVRI 2013) and determines a minimal model consisting of "fundamental functionality necessary to describe a functional environmental research infrastructure" (Chen et al. 2013b). Data collection is an example for a functionality of the data acquisition subsystem. Its role is to collect data, typically in form of digital values, from instruments, and pre-process data to associate timestamps and metadata. Data discovery and access is an example for a functionality of the data access subsystem. Its role is to retrieve, using appropriate search technologies, requested data from data resources of the infrastructure.

Reference model viewpoints

ENVRI-RM is defined from three viewpoints: science, information, and computational. Figure 2 represents the ENVRI-RM viewpoints and relations graphically. Each viewpoints describes environmental research infrastructure from a perspective. The three viewpoints are linked and together define the reference model.

The *science viewpoint* "intends to capture the requirements for an environmental research infrastructure from the perspective of the people who perform their tasks and achieve their goals as mediated by the infrastructure" (ENVRI 2013). (Unless stated otherwise, this paper quotes ENVRI 2013.) A broad categorization of people associated with environmental research infrastructure considers scientists who use the infrastructure; technicians who build, maintain, and operate the infrastructure; and managers who govern and administer the infrastructure.

The viewpoint defines five communities: data acquisition, data curation, data publication, data service provision, and data usage. Each community is described for sets of relevant community roles and behaviours. Communities interact with subsystems. A particular agent, for instance a scientist, can be a member of several communities.

A role in a community "is a prescribing behaviour that can be performed any number of times concurrently or successively." Roles can be active or passive. Active roles are associated with human agents. Passive roles are associated with non-human agents. For example, sensor and observer are roles in the data acquisition community. Sensor is a passive role and observer is an active role.

A behaviour of a community "is a composition of actions performed by roles normally addressing separate [research activity] requirements." For example, data collection is a

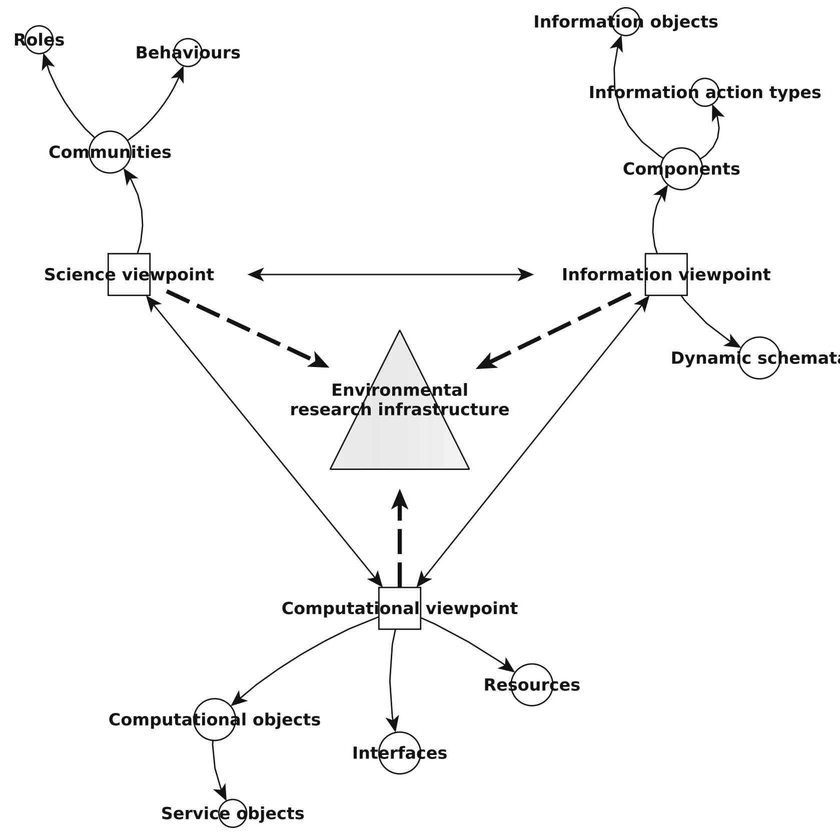


Fig. 2 The ENVRI-RM viewpoints with the main aspects described by the reference model

behaviour of the data acquisition community. It is performed by a data collector, an active role. The composition of actions of the behaviour includes (1) obtaining digital values from a sensor or an observer and (2) associate consistent timestamps and necessary metadata.

The *information viewpoint* provides "a common abstract model for the shared information handled by the infrastructure." Furthermore, it "describes how the state of the data evolves as [a result] of computational operations" and "defines the constraints on the data and the rules governing the processing of such data."

The reference model presents various aspects of the information viewpoint, notably components and dynamic schemata. The components aspect defines, in particular, collections for information objects and information action types. Information objects model information entities manipulated by the system. Information actions types model information processing in the system. Actions are associated with information objects. Objects are participants in actions. As an illustrative example, a measurement result is an information object, namely data processed by the system. Check quality is an information action type used to verify the quality of data, and is associated with the measurement result information object. Thus, measurement result participates in the check quality action type.

Dynamic schemata, a second aspect of the information viewpoint, "specify how the information evolves as the system operates." The evolution of information begins with specifications for the design of measurement (which includes the scientific question) and the method of measurement (which includes site selection). The execution of measurement generates measurement result information objects. They are stored as persistent data, and subsequently enriched, reviewed, published, queried, processed. Thus, dynamic schemata specify a system's data life-cycle. Life-cycle steps can be traced to obtain data provenance information about used objects, produced objects, and applied actions.

The *computational viewpoint* specifies "the major computational objects expected within an [environmental] research infrastructure and the interfaces by which they can be interacted with." The 'archetypical' environmental research infrastructure has a brokered service-oriented architecture. Functionality is thus encapsulated by service objects. Service objects control resources, and access to service objects is managed by brokers. Computational objects encapsulate functionality, which is exposed through interfaces. They exchange messages through operation interfaces (server or client) and deliver data through stream interfaces (producer and consumer).

Computational objects are organized along the five subsystems. Users interact with an environmental research infrastructure via a scientific gateway, a computational object of the community support subsystem. A Web community portal is a typical scientific gateway. Scientific gateways deploy virtual laboratories, which mediate interaction between particular groups of users and infrastructure subsystems. As an example for a particular type of virtual laboratory, experiment laboratories mediate interaction between curated data and data processing facilities. They support researchers in the deployment of datasets for processing and in the acquisition of results from computational experimentation.

Data acquisition is handled by instrument controllers. Instrument controllers encapsulate functionality of instruments or, more generally, (raw) data sources, and are managed by acquisition services. Acquisition services ensure the policy-conforming delivery of data into the infrastructure.

Data curation is handled by data store controllers. Data store controllers encapsulate functionality for the persistence and management of datasets, and are managed by curation services. The data transfer service, a type of curation service, provides data transporters to manage the movement of data. For instance, raw data collectors manage the movement of data from instruments to data stores, i.e. from the data acquisition subsystem to the data curation subsystem.

Data access is handled by data brokers. Data brokers act as intermediaries for access to data of the data curation subsystem. Brokers verify and validate access requests. Validated requests are forwarded to the relevant data curation service.

Data processing is handled by process controllers. Process controllers represent computational functionality of execution resources, and are managed by coordination services. Coordination services delegate processing tasks registered by the data processing subsystem to execution resources, coordinate workflows, and initiate task execution.

In addition to describing the computational objects and interactions relevant to subsystems, the ENVRI-RM also describes interactions between computational objects of different subsystems. For instance, raw data collection relies on coordination between the data acquisition subsystem and the data curation subsystem, in particular between acquisition service and data transfer service computational objects.

Reference model extension

ENVRI-RM focuses on data, their acquisition, curation, access, and processing. We extend ENVRI-RM to model the acquisition of knowledge from data, and the curation, access, and processing of knowledge. The ENVRI-RM *extension* is itself a model and is called +K, which stands for *plus knowledge*. The extension is inspired by ENVRI-RM, in the sense that it reuses some of the modelling choices made for ENVRI-RM. Furthermore, it can be superimposed on ENVRI-RM. The result of such superimposition is the ENVRI-RM+K model.

Extension subsystems

The +K extension introduces four subsystems: knowledge acquisition, knowledge curation, knowledge access, and knowledge processing. Figure 3 illustrates the four +K subsystems.

The knowledge acquisition subsystem acquires knowledge from data. Knowledge acquisition is a process and consists of three sub processes: information attainment, information mapping, and knowledge composition. Information is attained from data. Attained information is mapped to atomic entities of a conceptual model. Mapped information is composed to structured entities of a conceptual model, i.e. composed knowledge. While a structured entity is knowledge, the atomic entities constituents of the structured entity are information. For instance, knowledge acquisition may attain a threshold t from data; map the threshold t to a rule atom a that requires the variable v to exceed t, i.e. $a : v > t$; and compose the rule atom a to a rule $a \wedge b \rightarrow c$, whereby $a \wedge b$ and c are the rule antecedent and rule consequent, respectively. The threshold t is attained information; the rule atom a is mapped information, an atomic entity of a conceptual model; and the rule is composed knowledge, a structured entity of a conceptual model.

Aamodt and Nygård (1995) proposed a definitional framework for the concepts of data, information, and knowledge within the context of an agent decision-making

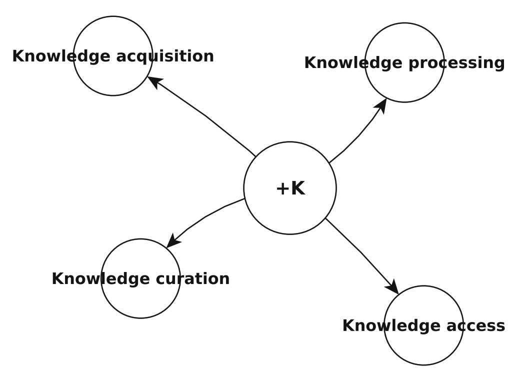


Fig. 3 The knowledge acquisition, knowledge curation, knowledge access, and knowledge processing subsystems of the +K extension to ENVRI-RM

process. The framework can be utilized to describe the differences between data, information, and knowledge in the ENVRI-RM+K model. According to Aamodt and Nygård, data are syntactic entities with no meaning; information is interpreted data, i.e. data with meaning; and knowledge is learned information, i.e. "information incorporated in an agent's reasoning resources," in other words information incorporated in an existing body of knowledge. Given sensor time-series data for the measured concentration of nutrients such as nitrogen and phosphorous in Finnish lakes, the sensor data are data, the threshold that distinguishes nutrient poor and nutrient rich lakes is information, and rules that classify a lake into nutrient poor or rich are knowledge. Aamodt and Nygård's Data-Information-Knowledge model includes three processes argued to be fundamental to "transforming data into information, deriving other information, and acquiring new knowledge." Interpretation is the process that transforms data into information. Elaboration is the process that derives new information. Learning is the process that acquires new knowledge. The interpretation process takes data as input, in particular sensor data, and returns *mapped* information as output. For instance, an interpretation process may utilize a clustering technique to obtain from data the threshold that distinguishes nutrient poor and nutrient rich lakes, and map the threshold to a rule atom. Interpretation is thus relevant to knowledge acquisition in the +K model. The elaboration process takes information as input and returns information as output. The process derives new information from existing information, i.e. derives atomic entities from existing atomic entities of a conceptual model, and is thus relevant to knowledge processing in the +K model. The learning process integrates "new information into an existing body of knowledge, in a way that makes it potentially useful for later decision making." The process is thus relevant to knowledge acquisition, as well as knowledge processing, in the +K model. The rule atom for the threshold is composed to a rule, and is integrated into an existing body of knowledge, which in the +K model is curated within the knowledge curation subsystem.

The knowledge curation subsystem facilitates quality control and preservation of knowledge. It also handles the representation of knowledge. The subsystem can curate various types of knowledge, specifically terminological and assertional knowledge. Terminological knowledge includes foundational knowledge, such as a the fact that sensing devices are sensors and thus physical objects, and domain knowledge, such as domain rules. In the +K model, assertional knowledge is primarily knowledge about a particular volume of reality monitored in space-time.

The knowledge access subsystem is concerned with the presentation and delivery of knowledge products. Retrieval of knowledge is enabled by query and search tools that support human or software agents in knowledge discovery. Knowledge discovery is achieved by inspecting knowledge or by following semantic relations. For example, knowledge can be inspected to discover defined sensing devices. Given a discovered sensing device, it is possible to follow semantic links to discover the property that is observed by the sensing device.

The knowledge processing subsystem handles knowledge reasoning, visualization, and analysis. The subsystem can include generalized and specialized services, e.g. for statistical analysis or visualization of knowledge. For instance, a knowledge processing subsystem may provide a service that visualizes what is known about a particular volume of reality along temporal and spatial dimensions.

Extension functionality

Each of the four +K subsystems for the acquisition, curation, access, and processing of knowledge addresses a range of concerns by implementing certain functionality. Figure 4 illustrates how +K functionality is partitioned amongst the four +K subsystems.

The *knowledge acquisition subsystem* is primarily concerned with the attainment of information from data, the mapping of attained information to atomic entities of a conceptual model, and the composition of mapped information to structured entities of a conceptual model.

There are three modes in which the knowledge acquisition subsystem can attain information from data: acquisition, extraction, and collaborative. In acquisition mode, it is human agents (actors) that attain information from data. In extraction mode, it is software agents that attain information from data. In collaborative mode, human and software agents attain information from data collaboratively. The distinguishing factor is in the procedure used to attain information from data, in particular the type of agents involved in the procedure. In extraction mode, the procedure is automated and involves software agents, exclusively. In acquisition mode, the procedure is not automated and involves human agents. In collaborative mode, the procedure is semi-automated and involves both human and software agents.

The procedure is limited to the task of attaining information from data and does not extend to related tasks that are typically necessary to design or implement (software) agents involved in the procedure. For instance, a software agent based on a supervised machine learning algorithm typically involves human agents in the design phase, e.g. in the preparation of a labelled dataset used to train the software agent. However, the (trained) software agent automates the task of attaining information from data. Information is thus attained in extraction mode. A software agent based on a supervised machine learning algorithm may also be in

Fig. 4 The +K functionality and its partitioning amongst the four +K subsystems

collaborative mode with human agents. For example, this is the case when it is known that the performance of the software agent is insufficient and its result needs to be reviewed by a human agent.

Resulting from the information attainment process is attained information. Attained information is, generally, a value of primitive data type, such as a number or a string. For instance, the label returned in machine learning classification is a value and is attained information. Attained information is subsequently mapped to atomic entities of a conceptual model. Resulting from the information mapping process is mapped information. For instance, attained information may be mapped to an individual, instance of a concept. The information attainment and information mapping processes are constituents of Aamodt and Nygård's data interpretation process. Mapped information is subsequently composed to structured entities of a conceptual model. Resulting from the knowledge composition process is composed knowledge. Generally, it is specifying relations between mapped information that results in composed knowledge. Composed knowledge is itself an individual, instance of a concept. The knowledge composition process integrates information into an existing body of knowledge and can thus be understood as Aamodt and Nygård's learning process.

The primary concerns of the *knowledge curation subsystem* are knowledge quality checking, knowledge storage and preservation, knowledge representation, and knowledge identification. Knowledge quality checking detects and corrects (or removes) inconsistent or inaccurate knowledge. It may be implemented by software agents, human agents, or collaboratively. Knowledge storage and preservation deposits (over the long-term) knowledge according to specified policies and makes knowledge accessible on request. It is supported by a knowledge store. Knowledge representation represents knowledge consistently with relevant conceptual models. It is supported by appropriate languages and technologies. Knowledge identification assigns global unique identifiers to knowledge.

The primary concerns of the *knowledge access subsystem* are knowledge discovery and retrieval as well as knowledge publication. Functionality for knowledge discovery and retrieval uses suitable search technology to retrieve requested knowledge from a knowledge resource. The knowledge resource is, generally, the system associated with the knowledge storage and preservation functionality, and can be a distributed system of knowledge stores. The knowledge publication functionality provides clean, well-annotated, anonymity-preserving knowledge in a suitable format, and by following specified publication and sharing policies. A publication policy may specify knowledge to be publicly accessible or restricted. Knowledge may be published for download, typically over the Web. Access may also be supported via (Web) Application Programming Interfaces (API).

The *knowledge processing subsystem* is concerned with various forms of knowledge processing, in particular knowledge visualization and analysis as well as different types of reasoning. Generally, assertional knowledge is located in time and space. Knowledge can thus be visualized along these two dimensions. Time lines and maps can support such visualization. Knowledge analysis generally depends on the domain and problem, which thus define the particular methods of interest in knowledge analysis. For instance, given knowledge for drivers travelling roads, statistical analysis can be used to compute summary statistics, such as mean driving speed and standard deviation. Given knowledge for the spatio-temporal location of drivers and knowledge for the spatio-temporal location of storms, reasoning can infer knowledge for spatio-temporal locations in which drivers are at higher risk, implied by storms and drivers that overlap in space-time.

Extension viewpoints

Following the approach used for ENVRI-RM, we define the +K extension from the science, information, and computational viewpoints.

Science viewpoint

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

The *knowledge acquisition community* is who attains information from data, maps attained information to atomic entities of a conceptual model, and composes mapped information to structured entities of a conceptual model.

Key *roles* in the knowledge acquisition community include the expert, attainer, mapper, and composer. Expert is an active role and is either a domain expert or a computer expert. Domain experts are scientists and researchers in earth and environmental science, or a related field of science. The domain expert is the primary source of domain knowledge. She defines the knowledge acquisition problems and provides contextual information relevant to knowledge acquisition. Computer experts implement, deploy, and maintain sensing devices and sensor networks as well as computer and software systems. The computer expert is the primary source of technical knowledge and is the person responsible for extending conceptual models and software in order to implement knowledge acquisition problems. Domain experts collaborate with computer experts. It is possible for an individual expert to be both a domain expert and a computer expert. Information is attained from data by an attainer, an active or a passive role exerted by an agent that attains information from data. Of primary interest here is the passive role of attainer, i.e. the extractor. Extractors are software agents that can attain information automatically or in collaboration with human agents. In collaborative mode, the extractor automatically attains information which is, however, subsequently revised by human agents—specifically, by a knowledge curator. The mapper and the composer are generally passive roles because these tasks can be automated. In the science viewpoint, the knowledge acquisition subsystem represents a passive role of the knowledge acquisition community.

Key *behaviours* of the knowledge acquisition community include method specifications, knowledge acquisition, conceptual model extension, and software extension. Knowledge acquisition consists of three behaviours: information attainment, information mapping, and knowledge composition. These behaviours are performed by the three roles attainer, mapper, and composer, respectively. Methods are specified for knowledge acquisition, and attainment, mapping, and composition in particular. The complexity of knowledge acquisition often requires interdisciplinary collaboration in method specification, involving both domain and computer experts. Conceptual model and software extension are behaviours performed by computer experts. Conceptual model extension includes the alignment of domain concepts and relations with those of foundational conceptual models. Software extension includes the design, implementation, testing, and deployment of software agents required for knowledge acquisition.

The *knowledge curation community* is who curates, maintains and archives knowledge. Key *roles* in the knowledge curation community include the knowledge curator, knowledge representer, knowledge identifier, and knowledge store. The knowledge curator is an active role, a human agent who verifies the quality of knowledge, preserves and maintains knowledge as a resource, and prepares various required knowledge products. The knowledge representer is a passive role, a software agent that uses languages and technologies to represent knowledge. The knowledge identifier is a passive role, a software agent that creates and assigns identifiers to knowledge. The knowledge store is a passive role, a software agent that persists and preserves knowledge. It is most obviously implemented by a knowledge base. In the science viewpoint, the knowledge curation subsystem represents a passive role of the knowledge curation community. Relevant roles in the knowledge curation community are also the storage administrator and the storage, defined as part of the ENVRI-RM data curation community.

Key *behaviours* in the knowledge curation community include knowledge quality checking, knowledge representation, knowledge identification, knowledge persistence, knowledge preservation, knowledge replication, knowledge product generation. Quality checking is typically performed by an active role, specifically a knowledge curator who detects and corrects (or removes) inconsistent or inaccurate knowledge. However, akin to data quality checking, software agents may quality check knowledge to some degree. The required degree of quality control largely depends on the performance of knowledge acquisition. Knowledge product generation is a behaviour performed by a knowledge curator who generates knowledge products against requirement specifications and standardized formats and descriptions. Knowledge representation, identification, and persistence are behaviours performed by corresponding roles (representer, identifier, and store, respectively). In practice, these behaviours may be performed by one or more software agents. Knowledge preservation is also concerned with the preservation of provenance information, in particular information about the agents and methods involved in knowledge acquisition and processing. Knowledge replication is a behaviour performed by a storage administrator who creates, deletes, and maintains the consistency of copies of knowledge sets on multiple storage devices.

The *knowledge publication community* is who assists knowledge publication, discovery and access. Key *roles* in the knowledge publication community include the knowledge publication repository and knowledge consumer. The knowledge publication repository is a passive role, a facility for the deposition of published knowledge. Knowledge can be published in various forms, e.g. files for download or an endpoint for query. Knowledge consumers are roles exerted by human or software agents that receive and use knowledge published by knowledge publication repositories. In the science viewpoint, the knowledge access subsystem represents a passive role of the knowledge publication community. A relevant role in the knowledge publication community is also the semantic mediator, defined as part of the ENVRI-RM data publication community.

Key *behaviours* in the knowledge publication community include knowledge publication and knowledge discovery and access. Knowledge publication is a behaviour performed by a knowledge publication repository that provides clean, well-annotated, anonymity preserving knowledge. Knowledge discovery and access is a behaviour enabled by a knowledge access subsystem that retrieves requested knowledge from a knowledge resource by using suitable search technology.

The *knowledge service provision community* is who provides various services, applications and software tools used to process knowledge. Key *roles* in the knowledge service provision community include the knowledge provider and software engineer. The knowledge provider is an active or a passive role, an agent that provides knowledge to be used, primarily in processing. The software engineer is an active role, a person who implements services, applications, and software tools for knowledge processing. Knowledge processing (e.g. in visualization, reasoning, or analysis) can require domain program logic. Hence the role of software engineers. In the science viewpoint, the knowledge processing subsystem represents a passive role of the knowledge service provision community. Relevant roles in the knowledge service provision community are also the service provider, service registry, capacity manager, and service consumer, defined as part of the ENVRI-RM data service provision community.

Software implementation and knowledge processing are *behaviours* of the knowledge service provision community. Software implementation is a behaviour performed by a software engineer. Knowledge processing is a behaviour performed by a service provider that processes knowledge. Other relevant behaviours, such as service registration and service composition, overlap with those of the ENVRI-RM data service provision community.

The *knowledge usage community* is who makes use of knowledge and service products, and transfers knowledge into understanding. The *roles* and behaviours in the knowledge usage community are the same as those of the ENVRI-RM data usage community. Relevant roles include the scientist or researcher, technologist or engineer, educator, policy or decision maker, industrialist or consultant, media, citizen, and the general public. Notably, the scientist in the knowledge usage community is a role different from the domain expert in the knowledge acquisition community. Compared to the former, the latter performs a role in knowledge acquisition, for instance in its specification. Naturally, an individual person may be a scientist in the knowledge usage community as well as a domain expert in the knowledge acquisition community.

Information viewpoint

The information viewpoint intends to provide an abstract model for the shared information objects that are relevant to the +K extension by specifying their types and relations between types. The information viewpoint discusses the following aspects of the extension: components, dynamic schemata, static schemata, subsystem schemata.

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

Information objects are defined to capture three types of information relevant to the extension. The first type of information captured by information objects, i.e. meta information of knowledge collections, includes specifications for knowledge acquisition, knowledge curation, knowledge access, and knowledge processing. Such specifications are documents and are created by experts. They result from collaboration between domain and computer experts. A specification for knowledge acquisition describes the type of data objects involved in, and the type of knowledge objects resulting from, knowledge acquisition. Such specification includes information about the methods used in information attainment, e.g. information about the algorithms, information for how attained information objects are mapped to atomic entities of a conceptual model, and information for how mapped information objects are composed to structured entities of a conceptual model. A specification for knowledge curation documents what is known about the quality of knowledge acquisition, e.g. the performance of involved algorithms, and thus the requirements for quality checking. The specification details the type of knowledge store, including information about its deployment, configuration, and connection. A specification for knowledge access describes how knowledge can be accessed and how knowledge is published. A specification for knowledge processing documents for what purpose knowledge is processed and how processing is achieved, in particular the involved (software) agents.

The second type of information captured by information objects, i.e. data and knowledge processed by the system, includes the types of data objects, attained and mapped information objects, and composed knowledge objects. Data objects are the input to knowledge acquisition. Attained information objects are the result of information attainment. Generally, attained information objects are values of some primitive data type. For instance, the label returned in machine learning classification is an attained information object. Attained information objects are mapped to atomic entities of a conceptual model. The result are mapped information objects. For instance, an attained information object can be mapped to an individual, instance of a class of the conceptual model. Finally, mapped information objects are composed to structured entities of a conceptual model. The result are composed knowledge objects. Composed knowledge generally consists of information attained by one or more extractors, mapped to atomic entities of a conceptual model, and structured by relations among mapped information.

The third type of information captured by information objects, i.e. information used for the management of knowledge, includes information for knowledge provenance used to trace state changes of knowledge in its life-cycle.

Information action types model how knowledge is processed in the system. Four action types are concerned with the creation of specifications for knowledge acquisition, curation, access, and processing. The most fundamental information action type is arguably to perform knowledge acquisition, which consists of three information action types—for information attainment and mapping, and knowledge composition. Another fundamental information action type is to represent composed knowledge, according to the syntax and semantics of knowledge representation languages. Various information action types are for the management of knowledge, in particular to store, check the quality, and query knowledge. Another important information action type is concerned with the processing of knowledge obtained from one or more knowledge resources. Each of these actions changes the state of knowledge objects. Some of the ENVRI information action types are also relevant to the extension. Examples include carry out backup, to replicate knowledge to an additional knowledge storage; assign unique identifier, to obtain a unique identifier and associate it to knowledge objects; build conceptual models, generally in form of domain concepts that extend foundational concepts.

Information objects may exist as multiple *instances*. One purpose of instances is to record knowledge state changes as effects of actions. For instance, a knowledge object resulting in knowledge acquisition is in state acquired, and as effect of the process knowledge action the knowledge object is in state processed. Various *knowledge states* are relevant to the extension. The state objects are in during knowledge acquisition can be recorded using the three states attained, mapped, and composed. Of particular interest is the knowledge state processed, of which inferred is a specialization.

Dynamic schemata Knowledge acquisition must be specified and implemented before it can be performed. At a minimum, the specification must detail what data objects are relevant to information attainment; the methods involved in information attainment; how attained information objects are mapped to atomic entities of a conceptual model; how mapped information objects are composed to structured entities of a conceptual model; how conceptual models are extended with relevant knowledge; and how software is extended in order to implement required program logic, in particular software agents that implement knowledge acquisition. Furthermore, conceptual models and, in particular, software extensions must be implemented.

When knowledge acquisition is implemented, it can be carried out. The +K extension expects data objects that can directly serve as input to information attainment, i.e. data objects that do not require further data processing prior to information attainment. In practice, measurement data, i.e. raw data collected from sensing devices, is seldom fit for information attainment. Hence, measurement data typically undergoes a series of data processing steps before the resulting data objects can serve as input to information attainment. For instance, measurement data for vibration in road pavement used to detect vehicles travelling a road section needs to be processed from time to frequency domain before vibration patterns can be classified (naturally, the details depend on the method).

The result of knowledge acquisition is composed knowledge. Handling of composed knowledge includes the store knowledge action. Stored knowledge can be accessed, quality assessed, published, and processed.

Information attainment is the first step in the knowledge life-cycle. Steps following information attainment are information mapping and knowledge composition. Composed knowledge objects are persisted. Persisted knowledge may be retrieved and processed, in particular reasoned. The steps in knowledge life-cycles can be represented as knowledge provenance.

Static schemata Static schemata specify the state of information objects at specific time points. Discussed are static schemata with constraints for knowledge acquisition specification, conceptual model and software extensions, and knowledge publication.

Constraints for knowledge acquisition specifications determine the preconditions required for knowledge acquisition implementation. The specification must record information relevant to information attainment, information mapping, and knowledge composition. For information attainment, the specification must describe the type of data object from which information is attained, the methods used in information attainment, and the resulting type of attained information object. Of interest is, in particular, information about the dataset (or stream of) data objects, method configuration and performance in information attainment, and information about the data type and semantics of attained information objects. For information mapping, the specification must describe the conceptual model and the method used to map attained information objects to atomic entities of the conceptual model. Of interest is, in particular, information about mapping rules and how mapped information objects are aligned to the conceptual model. For knowledge composition, the specification must describe the method used to compose mapped information to structured entities of the conceptual model. Of interest is, in particular, information about composition rules that govern how mapped information objects become composed knowledge objects. Additionally, the specification must record how knowledge acquisition is implemented, tested, deployed, and executed. Of interest are, in particular, the programming languages, unit tests, and deployment platform. Knowledge acquisition must be implemented, tested, and deployed according to the specification for knowledge acquisition.

Constraints for conceptual model and software extensions specify the preconditions required for conceptual model and software adoption into production. Conceptual models must be validated before adoption. This typically includes checking their consistency. Inconsistent conceptual models must be revised. Similar constraints apply to software extensions, which at a minimum must be tested using appropriate unit tests, prior to deployment.

Constraints for knowledge publication specify the preconditions required for preparing knowledge to be (publicly) accessible. In particular, the constraints require that accessible knowledge is mapped to entities of a conceptual model, and that the conceptual model is accessible. Knowledge publication also requires that a decision is made that knowledge can be seen by the public, or by a more restricted user community.

Subsystem schemata Subsystem schemata organize information objects and information action types amongst the +K subsystems.

Within the knowledge acquisition subsystem, the specification for knowledge acquisition is the information object that results from the specify knowledge acquisition information action type. Knowledge acquisition is performed on data objects. Performed are in particular information attainment, information mapping, and knowledge composition. Each of these information action types results in a corresponding information object (the attained, mapped, and composed information or knowledge objects, respectively). The result of the perform knowledge acquisition action are composed knowledge objects.

Within the knowledge curation subsystem, the specification for knowledge curation is the information object that results from the specify knowledge curation information action type. The store knowledge information action type stores knowledge objects to a knowledge store. Persisted knowledge can be quality checked.

Within the knowledge access subsystem, the specification for knowledge access is the information object that results from the specify knowledge access information action type. Knowledge is accessible as a result of the publish knowledge or the query knowledge information action types.

Finally, within the knowledge processing subsystem, the specification for knowledge processing is the information object that results from the specify knowledge processing information action type. Knowledge is processed as a result of the process knowledge information action type.

The track provenance information action type is applicable to all subsystems. The result of the action is a knowledge provenance object. Provenance is particularly relevant to the knowledge acquisition and knowledge processing subsystems.

Computational viewpoint

The computational viewpoint describes the computational objects of the +K extension and computational object interfaces. The computational objects of the extension are organized according to the four +K subsystems. Following ENVRI-RM, it is considered that the extension has a brokered, service-oriented architecture. Service objects encapsulate functionality and control resources. Brokers oversee access to services and validate requests. Brokers may also provide an interoperability layer to facilitate interaction between heterogeneous components.

The *knowledge acquisition subsystem* provides functionality for attaining information from data objects, mapping attained information to atomic entities of a conceptual model, and composing mapped information to structured entities of a conceptual model. Computationally, knowledge acquisition is described as sets of information attainers, information mappers, and knowledge composers associated with knowledge acquisition controllers.

A knowledge acquisition controller receives and directs data objects to attainers, attained information to mappers, and mapped information to composers. Thus, the controller orchestrates knowledge acquisition implemented by sets of computational objects. Data objects can be pulled by, or streamed to, knowledge acquisition controllers. A knowledge acquisition controller returns composed knowledge. Composed knowledge is subsequently forwarded to the knowledge curation subsystem. A knowledge acquisition service supports the registration, deregistration, and execution of knowledge acquisition controllers.

A knowledge attainer receives data objects and attains information. Its output are attained information objects. Generally, information is attained by means of computational models, e.g. data-driven or physically-based models. Computational models take the role of extractors. However, a human agent may also attain information. An information mapper receives attained information objects and returns mapped information objects. A knowledge composer receives mapped information objects and returns composed knowledge objects.

The *knowledge curation subsystem* provides functionality to persist and preserve knowledge objects. Computationally, knowledge curation is handled by a set of knowledge store controllers monitored and managed by a set of knowledge curation services, specifically knowledge annotation services and knowledge transfer services.

A knowledge annotation service implements the functionality required to annotate knowledge objects. The service is primarily intended for use in knowledge quality checking to update inconsistent or inaccurate knowledge. A knowledge transfer service supports the registration, deregistration, and execution of knowledge transporters, such as knowledge collectors, importers, and exporters. Knowledge objects can be pulled by, or streamed to, knowledge transporters, which associate with knowledge store controllers to persist knowledge objects. A knowledge collector receives knowledge objects returned by one or more knowledge acquisition services and directs knowledge objects to associated knowledge store controllers. Knowledge importers and knowledge exporters are intended for import and export of knowledge objects from or to an external destination. Knowledge objects are streamed to, or retrieved from, knowledge store controllers.

The *knowledge access subsystem* provides knowledge brokers that act as intermediaries for access to knowledge objects managed by the knowledge curation subsystem. Knowledge brokers intercede between the knowledge access subsystem and the knowledge curation subsystem. They implement the functionality required to negotiate knowledge object transfer and requests directed at knowledge curation services on behalf of agents. Knowledge brokers are responsible for verifying the agents making access requests and for validating those requests prior to directing them to the relevant knowledge curation service.

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

Discussion

We have briefly summarized the ENVRI *Reference Model*, ENVRI-RM. It is a model for the ‘archetypical’ environmental research infrastructure, distilled through the analysis of several concrete infrastructures. We have highlighted that the model is primarily concerned with data, data acquired in measurement or observation, and the curation, access, and processing of data.

We developed an *extension* to ENVRI-RM for the acquisition of knowledge from (processed) data, and the curation, access, and processing of knowledge. We called the extension +K, and argued that it may be superimposed on ENVRI-RM to form ENVRI-RM+K, a model for the 'archetypical' environmental research infrastructure that goes beyond data to include processes that acquire knowledge from data as well as processes for the representation, persistence, retrieval, analysis, visualization, inference of knowledge.

The purpose of this paper, and our main contribution, is to propose a possible model for a knowledge extension to ENVRI-RM. It is a contribution to ongoing discussions within a community of researchers who aim at a practical and clear architecture for environmental research infrastructure. We raise the question, and suggest a possible approach, for what happens to knowledge that is obtained in processing data curated by an environmental research infrastructure.

ENVRI-RM includes data mining as a functionality of the data processing subsystem. It also states that the data service provision community is "who provides various services, applications and software/tools to link and recombine data and information in order to derive knowledge." However, the model does not specify how that which results from data mining, i.e. derived knowledge, ought to be managed by an environmental research infrastructure, if at all. Our claim is that, beyond data, an environmental research infrastructure can, and perhaps should, manage derived knowledge. An infrastructure that does manage derived knowledge is a knowledge-based environmental research infrastructure.

To evaluate the implications of managing knowledge derived from data in a knowledge-based environmental research infrastructure is arguably an interesting research topic in its own right. We shortly discuss two implications that in our opinion are particularly interesting. First, the knowledge-based environmental research infrastructure acts as an integrated repository and processing infrastructure for scientific knowledge derived from data. Second, the explicit representation of derived knowledge, i.e. machine interpretable knowledge, enables the knowledge-based environmental research infrastructure to use existing knowledge and automatically perform computations on knowledge to derive new knowledge.

ENVRI-RM underscores the importance of an integrated repository and processing infrastructure for scientific data. The infrastructure acts as a resource which the scientific community can access to retrieve integrated quality scientific data. The community utilizes such data in scientific studies. The result of such studies is information, knowledge, and understanding. Such results are generally communicated through scientific articles or other documents, using natural language, figures, tables, or other means. As a result, information and knowledge is dispersed and represented heterogeneously. ENVRI-RM is a model for infrastructure that addresses the problem of dispersed scientific data of heterogeneous format and quality. The +K extension to ENVRI-RM is a model for infrastructure that addresses the problem of dispersed scientific knowledge of heterogeneous format and quality. Furthermore, by representing derived knowledge explicitly using methods in knowledge representation and reasoning the knowledge-based environmental research infrastructure "has knowledge – it's 'own' knowledge – and ways of processing that knowledge" (Aamodt and Nygård 1995). The infrastructure is thus equipped with functionality that not just supports retrieval and discovery of knowledge; it is also equipped with functionality that can automatically process knowledge, including the deductive inference of implicit knowledge.

The proposed +K extension does not address certain important aspects. First, it does not determine technological aspects of knowledge representation and reasoning, knowledge representation languages, as well as related technologies for the management, retrieval, and inference of knowledge. Second, the +K extension does not specify the types of knowledge that are of concern to the model. We discuss these two aspects in more details next and highlight some concrete technological choices and knowledge types of interest. We underscore that concrete implementations of the +K extension are not limited to the technologies discussed here and may opt for alternatives.

The suite of technologies developed within the W3C Semantic Web Activity[5] (Berners-Lee et al. 2001) can arguably address knowledge representation and, to some extent, reasoning in knowledge-based environmental research infrastructure. The Resource Description Framework (RDF) is a flexible graph (meta-)data model (Lassila and Swick 1999; Cyganiak et al. 2014b). Its basic construct, the RDF statement, is a structure consisting of a predicate that relates a subject to an object. Subjects and objects are resources, either an Internationalized Resource Identifier (IRI) (Dürst and Suignard 2005) or, in the case of objects, a (typed) literal, such as a string or an integer. A set of RDF statements forms an RDF graph, most intuitively visualized by directed arcs for the predicates relating subjects with objects. RDF was developed as a metadata model for Web resources, such as a Web page that is the subject about which RDF statements state something (e.g. its author). However, RDF resources are more general and include anything to which an IRI can be assigned, including physical things, such as a sensing device, or abstract concepts, such as ambient air. We can thus use RDF to state what is known about a sensing device, such as the observed property, or a particular volume of ambient air, such as temperature and relative humidity. In concrete +K implementations, RDF can

[5] http://www.w3.org/2001/sw/

serve as data model for knowledge acquired from processed data.

RDF Schema (RDFS) (Brickley and Guha 2004; 2014) and the Web Ontology Language (OWL) (W3C OWL Working Group 2012; 2009) are languages for the description of the concepts and relations of conceptual models. Descriptions have formal semantics and are axioms of an ontology (Gruber 1993; Baader et al. 2007). RDFS introduces the basic building blocks of a language for constructing ontologies, such as constructs for grouping RDF resources into classes or building class hierarchies. OWL builds on RDFS and introduces further language constructs, such as those for stating class equivalence or disjointness and for complex class descriptions. OWL is thus more expressive than RDFS. In concrete +K implementations, RDFS and OWL can support the construction of ontologies. Knowledge acquired from processed data is represented conformant to selected ontologies. There exist specialized ontologies for domains of interest to knowledge-based environmental research infrastructure, for instance the Semantic Sensor Network ontology (Compton et al. 2012), the RDF Data Cube Vocabulary (Cyganiak et al. 2014a), GeoSPARQL (Perry and Herring 2012), OWL-Time (Hobbs and Pan 2006), and the PROV-O provenance ontology (Lebo et al. 2013).

Of interest to +K implementations that adopt RDF, RDFS, and OWL are also RDF databases for the curation of knowledge, SPARQL (Prud'hommeaux and Seaborne 2008) endpoints for access to knowledge, and OWL reasoners and rule languages for knowledge inference. There is a plethora of software available for knowledge curation, access, and inference. Examples include the Stardog RDF database,[6] Apache Jena,[7] Profium Sense,[8] Sesame,[9] Pellet (Sirin et al. 2007),[10] HermiT (Shearer et al. 2008),[11] and the Semantic Web Rule Language (Horrocks et al. 2004).

The second aspect we shortly discuss is that of knowledge *types* of concern to the +K extension. In other words, knowledge about what is, concretely, acquired from data? Of particular interest is arguably knowledge about environmental phenomena observed in monitored volumes of space-time. In other words, knowledge about a monitored part of reality. In Situation Theory (Barwise and Perry 1980; 1981; Devlin 1991) a (structured) part of reality is called a situation. The concept of situation, as developed in Situation Theory and formalized in OWL by the Situation Theory Ontology (Kokar et al. 2009), can thus be useful for the representation of knowledge about monitored environmental phenomena. Knowledge is then, specifically, situational knowledge. Applications in this context have been developed by Stocker et al. (2014b) for vehicles detected and classified in road-pavement vibration data; Stocker et al. (2014a) for atmospheric new particle formation detected and classified in data for particle size distribution of polydisperse aerosols; (Stocker et al., Plant disease pressure situation modelling in agriculture Computers and Electronics in Agriculture, in review) for plant disease pressure in agriculture computed from weather data using a physically-based model; Clemente et al. (2013) for collision avoidance of ships in harbour areas; Fenza et al. (2010) for airport security; De Maio et al. (2012) for intrusion detection in a video-surveilled area of a bank; Doulaverakis et al. (2011) for security and surveillance. Relevant in this context are also theories of situation awareness (Endsley 1995; Stanton et al. 2006).

Naturally, knowledge about environmental phenomena observed in monitored volumes of space-time can be formalized with concepts other than that of situation, as described, in particular, in Situation Theory. A possible alternative to the concept of situation is the concept of event, which has been used in conjunction with Complex Event Processing techniques (Luckham 2002) in systems aimed at the detection of events from sensor data (Taylor and Leidinger 2011; Llaves and Kuhn 2014). An analysis that contrasts the concepts of situation and event is beyond the scope of this paper. However, we highlight Riker (1957) who called situations "the boundaries of events" and events the action occurring between situations, which seems to suggest that situations are snapshots and events make situations transition into new situations.

An (OWL) ontology is a knowledge base composed of a terminological box and an assertional box (Baader et al. 2007). The terminological box consists of axioms that describe the concepts and relations of a particular domain. For instance, the terminological box of the Semantic Sensor Network ontology specifies that a sensing device is a subclass of the sensor and device classes. The terminological box may also contain rules relevant to the domain. In contrast, the assertional box consists of assertions, specifically concept and role assertions. Concept assertions state the class membership of individuals. For instance, a concept assertion may state that a particular thermometer is an individual instance of the (SSN) sensing device class. Role assertions state role relationships between two individuals. For instance, a role assertion may state that a particular thermometer observes temperature, which is a particular property.

[6] http://stardog.com
[7] http://jena.apache.org
[8] http://www.profium.com/en/technologies/profium-sense
[9] http://rdf4j.org/
[10] http://clarkparsia.com/pellet/
[11] http://hermit-reasoner.com/

Knowledge acquired from processed data curated by knowledge-based environmental research infrastructure may be either terminological or assertional. In the particular case of situational knowledge, acquired knowledge is generally assertional, i.e. assertions about individual situations observed in volumes of space-time. However, a knowledge-based environmental research infrastructure may also acquire terminological knowledge from data, such as the threshold value for an atom of a rule. This was exemplified, e.g., by Stocker et al. (2011) who learn from data the threshold value for rules that classify lakes according to nutrient status.

In addition to knowledge about environmental phenomena observed in monitored volumes of space-time, of interest may also be knowledge about the environmental research infrastructure. Technicians may for instance be interested in the state of sensors; storage administrators require information about the state of the storage; engineers need to monitor the IT capacity of the infrastructure. Thus, knowledge about the environmental research infrastructure, and its computational objects in particular, is relevant to several roles of the ENVRI-RM+K model. Some of the required knowledge may be acquired from data curated by the infrastructure. However, of primary interest to the +K extension is arguably knowledge about environmental phenomena observed in monitored volumes of space-time, rather than knowledge about the environmental research infrastructure. This focus is certainly evident at least in our related work.

Case study

In this section we describe how the presented +K model applies to a case. The case is for the acquisition of situational knowledge for atmospheric new particle formation, from data for particle size distribution of polydisperse aerosols (Stocker et al. 2013, 2014a). We highlight the +K subsystems and functionality most relevant to the case, and discuss the case from the three +K viewpoints: science, information, and computational. As it clearly involves an environmental research infrastructure with a science community, it is arguably the most suitable case among the three cases we have developed so far. The other two are in intelligent transportation systems (Stocker et al. 2012, 2014b) and agricultural science (Stocker et al., Plant disease pressure situation modelling in agriculture Computers and Electronics in Agriculture, in review).

During new particle formation events, newly formed nano-sized particles grow through condensation and coagulation processes (Kulmala et al. 2004). Aerosol scientists study these atmospheric events for the impact of the resulting larger particles on climate and human health. The study of the events includes their identification and characterization, in analysis of measurement data for particle size distribution of polydisperse aerosols.

The case employs Situation Theory and Semantic Web technologies. Identified and characterized new particle formation events are thus modelled as situations. Situational knowledge is acquired from data using machine learning classification and is represented using relevant technologies and ontologies, in particular the Situation Theory Ontology. The task for classifiers is to determine whether a new particle formation event occurred during a particular day (classes E and NE, for the identification task) and, had an event occurred, to determine the clarity of the event as either strong, intermediate, or weak (classes 1, 2, and 3, respectively, for the characterization task).

Subsystems

The most relevant +K subsystems are knowledge acquisition and knowledge curation. Stocker et al. (2014a) also discuss forms of situational knowledge processing. In contrast, the description here is limited to knowledge acquisition and curation. In knowledge acquisition, relevant functionality is information attainment, information mapping, and knowledge composition. This process involves machine learning classification of data to attain information, and the composition of information mapped to atomic entities of the Situation Theory Ontology to situational knowledge. In knowledge curation, relevant functionality is knowledge identification, knowledge representation, and knowledge storage. Knowledge representation is by means of Semantic Web technologies, in particular the Web Ontology Language. Knowledge storage is enabled by a third-party knowledge base.

Science viewpoint

We describe the communities that are relevant to the case study, in particular the knowledge acquisition and knowledge curation communities. Table 1 provides a schematic overview.

Key roles in the knowledge acquisition community are those of aerosol scientist and software engineer. Aerosol scientists are domain experts and define new particle formation identification and characterization from data for particle size distribution to be a knowledge acquisition problem. They provide contextual information about new particle formation as well as access to data for particle size distribution as measured at particular locations over time. Software engineers are computer experts who interact with aerosol scientists to extend conceptual models with domain knowledge and develop software artifacts that implement domain program logic. Other relevant roles in the knowledge acquisition community are the attainer, mapper, and composer.

Table 1 Communities, roles, and behaviours of the science viewpoint that are of primary interest to the case study

Communities			
Knowledge acquisition		Knowledge curation	
Roles	Behaviours	Roles	Behaviours
Aerosol scientist	Method specification for knowledge acquisition	Knowledge representer	Knowledge identification
Software engineer	Conceptual model extension	Knowledge identifier	Knowledge representation
	Software extension	Knowledge store	Knowledge persistence
Extractor	Knowledge acquisition		Knowledge preservation
Mapper			
Composer			

The attainer is, specifically, an extractor. The three roles are passive, associated with software agents.

Key behaviours in the knowledge acquisition community include method specification for knowledge acquisition, conceptual model and software extension, and knowledge acquisition. Method specification for knowledge acquisition is a behaviour performed by aerosol scientists in collaboration with software engineers. Conceptual model and software extension are behaviours performed by software engineers. Conceptual model extension includes, for instance, the concept assertion that states that `npf` (for 'new particle formation') is an instance of the class `Relation`, as defined by the Situation Theory Ontology.

Key roles in the knowledge curation community are the knowledge representer, knowledge identifier, and knowledge store. The three roles are passive, associated with software agents. The knowledge representer uses relevant OWL ontologies and RDF to represent situational knowledge for new particle formation events. Knowledge is identified by means of IRIs assigned to RDF resources. The knowledge store persists and preserves situational knowledge and is implemented by the Stardog RDF database.

Information viewpoint

We describe the information objects and information action types that are relevant to the case study, and the dynamic schemata for knowledge acquisition, representation, and persistence. Table 2 provides a schematic overview.

Knowledge acquisition is specified and implemented before it is performed. Thus, we first determine what data objects are relevant to information attainment; the methods involved in information attainment; how attained information objects are mapped to atomic entities of a conceptual model; how mapped information objects are composed to structured entities of a conceptual model; how conceptual models are extended with relevant domain knowledge; and how software is extended to implement required program logic. We then implement and test the specified conceptual model and software extensions.

Knowledge is acquired from dataset observations. A dataset observation is an information object, more accurately an instance of a type of data object with semantics conformant to the concept `Observation` as defined by the RDF Data Cube Vocabulary. Composed knowledge are situations. A situation is an information object, more accurately an instance of a type of knowledge object with semantics conformat to the concept `Situation` as defined by the Situation Theory Ontology.

Information is attained by classifying dataset observations using Multi-Layer Perceptron (MLP) artificial neural networks, which are trained to identify and characterize new particle formation events. Dataset observations are vectors, and result in processing daily data for measured particle size distribution. The daily data matrix is processed to a daily vector using Singular Value Decomposition.

The perform information attainment information action type classifies dataset observations. This results in an

Table 2 Information objects and information action types of the information viewpoint that are of primary interest to the case study

Information objects	Information action types
Specification for knowledge acquisition	Perform information attainment
Dataset observation	Perform information mapping
Attained information object	Perform knowledge composition
Mapped information object	Represent knowledge
Situation	Store knowledge

Table 3 Computational objects of the computational viewpoint that are of primary interest to the case study

Computational objects	Input	Output
Situation engine	Dataset observations	Situations
Learning module	Dataset observations	Situations
Information attainer	Dataset observations	Attained information
Information mapper	Attained information	Mapped information
Knowledge composer	Mapped information	Situations
Situation writer	Situations	-
Store module	Situations	Situations
Knowledge base	Situations	Situations

attained information object, i.e. a value in {NE, 1, 2, 3}. For values in {1, 2, 3}, the perform information mapping information action type maps the attained information object to an atomic entity of a conceptual model, specifically an individual instance of the concept `Value` as defined by the Situation Theory Ontology. In addition, the value for the temporal location (i.e. the day) provided by the dataset observation is also mapped to an individual, instance of the concept `Interval` as defined by OWL-Time.

The perform knowledge composition information action type composes the information objects c and t for the event class and temporal location, respectively; the individual `npf`, instance of the class `Relation`; an individual instance of the class `Situation` and an individual instance of the class `ElementaryInfon` as defined by the Situation Theory Ontology, both created at this stage, to situational knowledge, i.e. a situation. Composition is achieved by relating individuals so as to state that the situation supports the (elementary) infon with objects c and t standing in the `npf`-relation. In plain English, the situation states that on day t an event of new particle formation with clarity c occurred.

The represent knowledge information action type represents situations according to relevant OWL ontologies in RDF. Finally, the store knowledge information action type persists situations to the knowledge base.

Computational viewpoint

We describe the computational objects that are relevant to the case study, in particular the situation engine, the situation writer, the learning module, the store module, and the knowledge base. Table 3 provides a schematic overview.

The situation engine is a computational object of the knowledge acquisition subsystem, more accurately a knowledge acquisition controller. It obtains dataset observations and returns situations. Its main purpose is to orchestrate situational knowledge acquisition from data using one or more learning modules. A learning module is a further computational object of the knowledge acquisition subsystem. It implements an information attainer, information mapper, and knowledge composer. The information attainer is backed by a computational model for information extraction from data. In the case presented here, we developed a situation engine with learning module that implements an information attainer backed by the trained machine learning models used to classify dataset observation, and designed to identify and characterize new particle formation.

The situation writer is a computational object of the knowledge curation subsystem, more accurately a knowledge collector that receives situations returned by one or more situation engines, and directs situations to associated store modules. The store module is a further computational object of the knowledge curation subsystem, more accurately a knowledge store controller. The store module encapsulates the interface required to interact with knowledge stores. The interface supports adding, updating, and removing situations. The knowledge store is implemented by the Stardog RDF database, which serves as a knowledge base for (situational) knowledge.

Conclusion

Analyzing the ENVRI Reference Model, ENVRI-RM, a model for the 'archetypical' environmental research infrastructure and a result of the ENVRI project, we have highlighted how ENVRI-RM is primarily concerned with data, its acquisition, curation, access, and processing. The model covers the data life-cycle in environmental research infrastructure.

With data mining and analysis, ENVRI-RM includes functionality for the processing of data aimed at obtaining knowledge. However, ENVRI-RM does not cover the knowledge life-cycle in environmental research infrastructure. In particular, ENVRI-RM does not detail knowledge acquisition, its curation, access, and processing.

We have presented an extension to ENVRI-RM for the acquisition of knowledge from processed data, and the curation, access, and processing of knowledge. We named the extension +K, which stands for *plus knowledge*. The extension aims at addressing the knowledge life-cycle in environmental research infrastructure and describes, in particular, in more details how knowledge is acquired from data by describing intermediate steps of information attainment and mapping, and knowledge composition. We argue that the +K model can be superimposed on ENVRI-RM to form the ENVRI-RM+K model. In fact, some of the ENVRI-RM model elements are directly useful to the +K extension. ENVRI-RM+K may be understood as a reference model for the 'archetypical' knowledge-based environmental research infrastructure.

To demonstrate the application of the +K model, we have discussed the three viewpoints for a concrete case study in which knowledge acquisition is for events of atmospheric new particle formation, and knowledge is acquired from data for particle size distribution of polydisperse aerosols.

This work contributes to ongoing discussions in research aimed at models and implementations of environmental research infrastructure. In particular, it aims at moving beyond data to consider how knowledge may be acquired from data and managed within an environmental research infrastructure. In discussing retrieval, discovery, and reasoning, the presented work also underscores some of the potential benefits of knowledge representation in environmental research infrastructure.

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

Aamodt A, Nygård M (1995) Different roles and mutual dependencies of data, information, and knowledge – An AI perspective on their integration. Data Knowledge Eng 16(3):191–222. doi:10.1016/0169-023X(95)00017-M. http://www.sciencedirect.com/science/article/pii/0169023X9500017M

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

Barwise J, Perry J (1980) The Situation Underground. In: Barwise J, Sag I (eds) Stanford Working Papers in Semantics vol 1, Stanford Cognitive Science Group, pp 1–55

Barwise J, Perry J (1981) Situations and Attitudes. The Journal of Philosophy 78(11):668–691. http://www.jstor.org/stable/2026578

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

Brickley D, Guha R (2004) RDF Vocabulary Description Language 1.0: RDF Schema. Recommendation, W3C. http://www.w3.org/TR/2004/REC-rdf-schema-20040210/

Brickley D, Guha R (2014) RDF Schema 1.1. Recommendation, W3C. http://www.w3.org/TR/2014/REC-rdf-schema-20140225/

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

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

Clemente S, Loia V, Veniero M (2013) Applying cognitive situation awareness to collision avoidance for harbour last-mile area safety. J Ambient Intell Humanized Comput: 1–5. doi:10.1007/s12652-013-0187-6

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 WD, Phuoc DL, Lefort L, Leggieri M, Neuhaus H, Nikolov A, Page K, Passant A, Sheth A, Taylor K (2012) The SSN ontology of the W3C semantic sensor network incubator group Web Semantics Science. Serv Agents World Wide Web 17(0):25–32. doi:10.1016/j.websem.2012.05.003

Cyganiak R, Reynolds D, Tennison J (2014a) The RDF Data Cube Vocabulary. Recommendation, W3C. http://www.w3.org/TR/2014/REC-vocab-data-cube-20140116/

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

Dal Maso M, Kulmala M, Riipinen I, Wagner R, Hussein T, Aalto P, Lehtinen K (2005) Formation and growth of fresh atmospheric aerosols: eight years of aerosol size distribution data from SMEAR II, Hyytiala, Finland. Boreal EnviroN Res 5:323–336

De Maio C, Fenza G, Furno D, Loia V (2012) Swarm-based semantic fuzzy reasoning for situation awareness computing. In: Fuzzy Systems (FUZZ-IEEE), 2012 IEEE International Conference on, pp 1–7. doi:10.1109/FUZZ-IEEE.2012.6251159

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

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

Dürst M, Suignard M (2005) Internationalized Resource Identifiers (IRIs). RFC 3987, IETF,. http://www.ietf.org/rfc/rfc3987.txt

Endsley MR (1995) Toward a theory of situation awareness in dynamic systems Human Factors. J Hum Factors Ergon Soc 37(1):32–64

ENVRI (2013) ENVRI Reference Model V1.1. Tech. Rep. RDTI-RI-283465, ENVRI. http://envri.eu/rm

Fenza G, Furno D, Loia V, Veniero M (2010) Agent-based Cognitive approach to Airport Security Situation Awareness. In: Proceedings of the 2010 International Conference on Complex Intelligent and Software Intensive Systems IEEE Computer Society, CISIS '10, pp 1057–1062

Gruber TR (1993) A translation approach to portable ontology specifications. Knowl Acquis 5(2):199–220. doi:10.1006/knac.1993.1008. http://www.sciencedirect.com/science/article/pii/S1042814383710083

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

Hart JK, Martinez K (2006) Environmental Sensor Networks A revolution in the earth system science? Earth Sci Rev 78(3-4):177–191. doi:10.1016/j.earscirev.2006.05.001

Hobbs JR, Pan F (2006) Time Ontology in OWL Working draft W3C. http://www.w3.org/TR/owl-time/

Horrocks I, Patel-Schneider PF, Boley H, Tabet S, Grosof B, Dean M (2004) SWRL: A Semantic Web Rule Language Combining OWL and RuleML. Tech. rep., W3C. http://www.w3.org/Submission/SWRL/

Junninen H, Lauri A, Keronen P, Aalto P (2009) Smart-SMEAR: online data exploration and visualization tool for SMEAR stations. Boreal Environ Res 14(4):447–457

Keller M, Schimel DS, Hargrove WW, Hoffman FM (2008) A continental strategy for the National Ecological Observatory Network. Front Ecol Environ 6(5):282–284. doi:10.1890/1540-9295(2008)6[282:ACSFTN]2.0.CO;2

Kokar MM, Matheus CJ, Baclawski K (2009) Ontology-based situation awareness. Inf Fusion 10(1):83–98

Kotamäki N, Thessler S, Koskiaho J, Hannukkala AO, Huitu H, Huttula T, Havento J, Järvenpää M (2009) Wireless in-situ Sensor Network for Agriculture and Water Monitoring on a River Basin Scale in Southern Finland: Evaluation from a Data User's Perspective. Sens 9(4):2862–2883. doi:10.3390/s90402862. http://www.mdpi.com/1424-8220/9/4/2862

Kratz TK, Arzberger P, Benson BJ, Chiu CY, Chiu K, Ding L, Fountain T, Hamilton D, Hanson PC, Hu YH, Lin FP, McMullen DF, Tilak S, Wu C (2006) Toward a Global Lake Ecological Observatory Network. Publ Karelian Inst 145:51–63

Kulmala M, Vehkamäki H, Petäjä T, Dal Maso M, Lauri A, Kerminen V, Birmili W, McMurry P (2004) Formation and growth rates of ultrafine atmospheric particles: a review of observations. J Aerosol Sci 35(2):143–176

Lassila O, Swick RR (1999) Resource Description Framework (RDF) Model and Syntax Specification Recommendation W3C. http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/

Lebo T, Sahoo S, McGuinness D (2013) PROV-O: The PROV Ontology W3C Recommendation W3C. http://www.w3.org/TR/prov-o/

Li W, Bhatia V, Cao K (2014) Intelligent polar cyberinfrastructure: enabling semantic search in geospatial metadata catalogue to support polar data discovery, Earth Science Informatics pp 1–13. doi:10.1007/s12145-014-0185-z

Llaves A, Kuhn W (2014) An event abstraction layer for the integration of geosensor data. International Journal of Geographical Information Science. doi:10.1080/13658816.2014.882513

Luckham DC (2002) The Power of Events, vol 204. Addison-Wesley Reading

Martinez K, Hart JK, Ong R (2004) Environ Sens Netw Comput 37(8):50–56. doi:10.1109/MC.2004.91

Perry M, Herring J (2012) OGC GeoSPARQL - A Geographic Query Language for RDF Data. Tech. Rep. OGC 11-052r4, Open Geospatial Consortium

Prud'hommeaux E, Seaborne A (2008) SPARQL Query Language for RDF Recommendation W3C. http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/

Riker WH (1957) Events and Situations. J Philos 3:57–70. http://www.jstor.org/stable/2022192

Shearer R, Motik B, Horrocks I (2008) HermiT: A highly-efficient OWL reasoner. In: Proceedings of the 5th International Workshop on OWL: Experiences and Directions (OWLED 2008), pp 26–27

Sirin E, Parsia B, Grau B, Kalyanpur A, Katz Y (2007) Pellet: A practical OWL-DL reasoner Web Semantics Science. Serv Agents World Wide Web 5(2):51–53. doi:10.1016/j.websem.2007.03.004

Stanton NA, Stewart R, Harris D, Houghton RJ, Baber C, McMaster R, Salmon P, Hoyle G, Walker G, Young MS, Linsell M, Dymott R, Green D (2006) Distributed situation awareness in dynamic systems: theoretical development and application of an ergonomics methodology. Ergon 49(12-13):1288–1311. doi:10.1080/00140130600612762. pMID: 17008257

Stocker M, Rönkkö M, Villa F, Kolehmainen M (2011) The Relevance of Measurement Data in Environmental Ontology Learning. In: Environmental Software Systems. Frameworks of eEnvironment, IFIP Advances in Information and Communication Technology, vol 359, Springer Boston, pp 445–453

Stocker M, Rönkkö M, Kolehmainen M (2012) Making Sense of Sensor Data Using Ontology: A Discussion for Road Vehicle Classification. In: Seppelt R, Voinov A, Lange S, Bankamp D (eds) International Congress on Environmental Modelling and Software, iEMSs, Leipzig, Germany, pp 2387–2394

Stocker M, Baranizadeh E, Hamed A, Rönkkö M, Virtanen A, Laaksonen A, Portin H, Komppula M, Kolehmainen M (2013) Acquisition and Representation of Knowledge for Atmospheric New Particle Formation. In: Hřebíček J, Schimak G, Kubásek M, Rizzoli AE (eds) Environmental Software Systems. Fostering Information Sharing, IFIP Advances in Information and Communication Technology, vol 413, Springer Berlin Heidelberg, pp 98–108. doi:10.1007/978-3-642-41151-9_10

Stocker M, Baranizadeh E, Portin H, Komppula M, Rönkkö M, Hamed A, Virtanen A, Lehtinen K, Laaksonen A, Kolehmainen M (2014a) Representing situational knowledge acquired from sensor data for atmospheric phenomena. Environ Model Softw 58(0):27–47. doi:10.1016/j.envsoft.2014.04.006. http://www.sciencedirect.com/science/article/pii/S1364815214001108

Stocker M, Rönkkö M, Kolehmainen M (2014b) Situational knowledge representation for traffic observed by a pavement vibration sensor network. IEEE Trans Intell Transp Syst 15(4):1441–1450. doi:10.1109/TITS.2013.2296697

Taylor K, 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, Pan J (eds) The Semantic Web: Research and Applications, Lecture Notes in Computer Science, vol 6644, Springer Berlin Heidelberg, pp 285–299. doi:10.1007/978-3-642-21064-8_20

W3C OWL Working Group (2009) OWL 2 Web Ontology Language Document Overview Recommendation W3C. http://www.w3.org/TR/2009/REC-owl2-overview-20091027/

W3C OWL Working Group (2012) OWL 2 Web Ontology Language Document Overview Recommendation W3C, Second Edition. http://www.w3.org/TR/2012/REC-owl2-overview-20121211/