

A Semantic IoT Early Warning System for Natural Environment Crisis Management

STEFAN POSLAD¹, (Member, IEEE), STUART E. MIDDLETON², (Senior Member, ACM),
FERNANDO CHAVES³, RAN TAO¹, OCAL NECMIOGLU⁴, AND ULRICH BÜGEL³

¹School of Electronic Engineering and Computer Science, Queen Mary University of London, London E1 4NS, U.K.

²Department of Electronics and Computer Science, University of Southampton IT Innovation Centre, Southampton SO16 7NS, U.K.

³Information and Knowledge Management Systems, Fraunhofer Institute of Optronics, System Technologies and Image Exploitation, Karlsruhe 76131, Germany

⁴Boğaziçi University, Istanbul 34342, Turkey

CORRESPONDING AUTHOR: S. POSLAD (stefan.poslad@qmul.ac.uk)

This work was supported in part by the European FP7 Funded Project TRIDEC under Grant 258723, in part by the other project partners in helping to deliver the complete project Syste, in particular, GFZ, and in part by the German Research Centre for Geosciences, Potsdam, Germany. The work of R. Tao was supported by the Queen Mary University of London for a Ph.D. studentship.

ABSTRACT An early warning system (EWS) is a core type of data driven Internet of Things (IoTs) system used for environment disaster risk and effect management. The potential benefits of using a semantic-type EWS include easier sensor and data source plug-and-play, simpler, richer, and more dynamic metadata-driven data analysis and easier service interoperability and orchestration. The challenges faced during practical deployments of semantic EWSs are the need for scalable time-sensitive data exchange and processing (especially involving heterogeneous data sources) and the need for resilience to changing ICT resource constraints in crisis zones. We present a novel IoT EWS system framework that addresses these challenges, based upon a multiseismic representation model. We use lightweight semantics for metadata to enhance rich sensor data acquisition. We use heavyweight semantics for top level W3C Web Ontology Language ontology models describing multileveled knowledge-bases and semantically driven decision support and workflow orchestration. This approach is validated through determining both system related metrics and a case study involving an advanced prototype system of the semantic EWS, integrated with a deployed EWS infrastructure.

INDEX TERMS Early warning system, Internet of Things, crisis management, semantic Web, scalable, time-critical, resilience.

I. INTRODUCTION

A. MOTIVATION AND CHALLENGES

Natural environment disasters may be caused by natural hazard events, such as tsunamis, or by manmade hazard events such as earth substrate drilling. These may in turn cause widespread natural environment damage that can take the affected regions years to recover from, following the onset of the disaster. An Early Warning System or EWS is a core type of IoT information system used for environment disaster risk and effect management. It helps prevent loss of life and reduces the economic and material impact of disasters [1]. In 2011, it has been estimated that the cost of installing an EWS for tsunami detection in the Indian Ocean was between \$30 to \$200 million dollars, depending on the number of sensor buoys used, the precision of the measurements; and that the benefit to cost ratio was 4:1, i.e., every dollar

spent on mitigation saved society four US dollars [2]. An EWS is distinct from other types of environment ICT monitoring systems in that it supports four main functions: Risk analysis of predefined hazards and vulnerabilities; Monitoring and warning by means of relevant parameters used for forecasts to generate accurate and timely warnings; Dissemination and communication of the risk information and warnings to those at risk; Response capability built upon response plans that leverage local capabilities and the preparation to react to warnings.

Typically, specific parts of natural environments are instrumented with fixed sensors to monitor them. These represent IoTs in the physical environment. Examples of such instrumented environments include drilling rigs, which actively alter the natural environment, and specific regions that are monitored because they are prone to potential

environment hazards, such as coastal regions where there is some risk that tsunamis may occur. This sensor data is then transmitted (upstream) to either an onsite, or remote, data processing centre, or to both when federated. These data centres run the (downstream) routine operational event detection, special event detection, event handling decision processes and command-control work-flows. Typical work-flows are pre-planned and include: Geographical Information System (GIS) processing to capture, store, analyse and present the spatial-temporal context of the environment as customised maps; sensor data-fusion processing, decision analysis and support for information alerts to authorities and citizens. These data exchanges tend to be synchronised, pre-determined and to use data structures that are pre-set by the command-control centre. The main requirements for physical environment IoT EWSs are:

1. *Time-critical sensor data exchange*, i.e., the combination of detection time, assessment time and citizen evacuation time needs to be minimal compared to the physical propagation time for a critical event, e.g., tsunami [3]. The seismic sensor sub-system of a tsunami EWS is expected to issue a warning within 2-3 minutes after an event is detected [4].
2. *To be able to scale-up (scalability)* to deal with information floods as publisher numbers and rates increase *and scale-down (resilience)* to handle local bottlenecks for upstream information communication caused by local physical network and power disruptions. Note it is presumed that the downstream communication is remote to, and away from, the region of the environment disaster. As such it is not as prone to be disrupted. It is also assumed to have some degree of fault-tolerance.
3. *An EWS needs to support semantics to support context-awareness of crisis events in order to adapt information services and to support data and service interoperability.*

Semantics refer to a representation that imparts meaning to concepts. *There are several potential benefits in using a semantic approach to design elements of an IoT EWS.* Semantics can promote richer knowledge-driven use of data. Semantics is able to define richer conceptualisations or models in terms of richer relationships between the model concepts. Concepts can represent devices such as sensors, or communication channels, data processing services, or work-flows and their data and processing contexts, e.g., a Tsunami buoy sensor is a specific type of sensor that supports all the general properties of a generic sensor. Thus, a semantic model can ease the way in which new types of sensor are plugged into the system through metadata driven automation.

Semantics can also lead to richer processing of these concepts using rule-based and logic inferencing, e.g., when the wave movement has a certain frequency range and exceeds a specific wave peak-to-trough threshold over a particular time, this triggers a potential tsunami data processing event. A semantic model to underpin service processes can also enhance service interoperability, orchestration and extension.

There are five main challenges when using a semantic approach:

1. *To specify what representation to use and where to use it*, i.e., it is usually not practical to generate and exchange semantic representations at the sensors.
2. *To specify which semantic concepts are required*, i.e., semantics can be introduced to enhance interoperability when fusing heterogeneous sensor datasets or used to select appropriate service work-flows for more flexible service orchestration.
3. *To define how any different domain standard semantic representations can be semantically mapped to each other* and linked to the raw data, and when this should practically occur.
4. *To adhere to any performance constraints when using semantics*, e.g., time-sensitivity, performance and resilience.
5. *The complexity in developing a usable shared semantic model*, hence, this is often developed iteratively.

In order to illustrate the use of a semantic model by an EWS, first, the use of a non-semantic model is considered. Typically raw data, formatted in binary, with no metadata, is published by the sensor hardware as these are very resource constrained and are designed to support efficient data transfer. A data client subscribed to the use of the sensor data would be expected to hard-code a shared knowledge of the sensor data structure into the client in order to parse it. An example of this would be to use netCDF (Network Common Data Form) formatted binary sensor data, exchanged using the AMQP (Advanced Message Queuing Protocol, see Section III.A) as its message payload. Although such binary data is quite efficient to exchange, it is more difficult to fuse with other heterogeneous sensor data, and it is difficult to query and process this data flexibly.

A semantic model includes explicit metadata and ontological concept definitions, e.g. domain measurement concepts like ‘water elevation’, so that clients can, if they want to, semantically map concepts and still understand the data they receive. An example of this is to use OGC’s O&M model and W3C’s SSNO ontology formatted as XML metadata, stored in a semantic registry, and associated with the data streams. The OGC O&M model, see Section III.B, is a simpler or lighter semantic model in the following sense: it defines concepts such as Features of Interest, Procedures, Observed Properties, etc. but defines only very basic relations (Object-Type Properties) between these concepts and few inference mechanisms (reasoning).

An example of a more complex, heavier, semantic model is the use of SSNO (see Section III.C). This was designed to allow richer modelling capabilities such as defined subclasses, constraints and, especially, the alignment with other existing domain and high-level ontologies, such as DUL, the set SWEET of ontologies, as well as the possibility to apply different levels of OWL reasoning. The main benefits of a “heavyweight” ontology is when data and information coming from different sources, including their corresponding metadata, is fused and combined and when this is used to infer new “knowledge”, independently of the up-stream (from the sensor) or downstream (from the knowledgebase) data. For example: upstream messages refer to single concepts or single

data “channels”. Sensors as raw data sources, upstream, do not make use of “relationships” between the different concepts or channels. Downstream, alert messages (in the tsunami scenario) as short semi-structured text message can be generated by means of heavyweight semantics, i.e. data fusion, simulations and/or other “inference” mechanisms by processing the stored sensor data.

B. SCOPE AND FOCUS

Although EWSs can be applied to several application domains, our focus is solely on their use to aid natural environment disaster management. As there are different types of natural disasters, we focus on a subset of these. In particular we focus on geologic hazards, rather than on atmospheric hazards, insect swarms, etc. Different types of hazards differ in the types of IoT they use in terms of sensors, sensor mobility, and how these communicate. We focus on fixed environment sensors, not mobile sensors, and not on remote sensors that have no direct contact with the natural environment, such as airborne sensors or satellites out in space. We also focus on rapid onset natural hazards whose primary effect takes of the order of several tens of minutes up to days to primarily affect a region, rather than on slow onset hazards such as droughts whose primary effect can take months to years to occur. We do not focus on humans as sensors who generate microblogs about crisis events in text and image format. Most disaster and emergency information systems are classified using the generic management functions they support: as decision support systems, expert systems (to guide novice users), database systems and document management systems (to organise data) or communication systems. They are not classified according to how the information model is structured, i.e., as a KMS or Knowledge Management System, or as a sub-type of KMS, i.e., as a semantic system to better enable some management function. Our focus here is the on the design and validation of semantic EWSs to support the EWS monitoring and warning function. Although, we developed and demonstrated a semantic EWS for use with two different types of natural hazard tsunami Natural Crisis Management (NCM) and Industrial Sub-surface Development (ISD), because of space limitations we emphasise the application to tsunamis (NCM) only here.

Our primary objective is to research and develop a semantic EWS for use in aiding management of rapid onset geological type natural disasters. Our second objective is to research and develop and validate a semantic EWS for such deployments. To the best of our knowledge, our novelty is that no current semantic EWS has been proposed and validated to meet these two objectives (see section II). Our third contribution is that based upon our design, implementation and validation experiences, *we highlight some of the key trends to advance the application of semantic computing to types of systems such as EWSs* (Section V).

The remainder of this paper is organised as follows. Related work is critically analysed (Section II). The experimental framework is discussed (Section III). The results and validation of the method are presented (Section IV). Finally, the conclusions are presented (Section V).

II. RELATED WORK

The semantic models used by EWSs in quick onset natural environment disaster situations are critically analysed and classified here. As EWSs tend to be quite specialised environment monitoring systems, the semantic models used by other types of natural environment ICT systems are also surveyed to assess whether or not their semantic models could be applied for EWS use.

A distinction is made between syntactical or structural representations, e.g., W3C XML extensions, versus representations with a richer explicit semantics (or meaning) such as W3C’s RDF (Resource Description Framework), RDF-S (RDF Schema) and OWL (Web Ontology Language). *Semantic representations can be viewed as a range of lightweight to heavyweight semantic conceptualisations [6]–[8], the range defined informally in terms of the expressivity of their semantic data structures.* Very lightweight ontologies provide the simplest model formalization for the task at hand to codify the meaning of nodes and their links e.g., they use tree-like structures where each node label is a language-independent propositional DL (Description Logic) formula [7]. Each node formula is subsumed by the formula of the node above. As a consequence, the backbone structure of a lightweight ontology is represented by subsumption relations between nodes. In addition to this, heavyweight ontologies use more complex formal logics to describe nodes, to inference and to prove theorems, e.g., OWL-DL or OWL-Full. EWS Semantics in practice are affected by time-sensitivity, scalability and resiliency, by local ICT resource constraints and by, a possibly temporary, lack of resource availability. The length of time the computation takes also affects its use as contexts change when resource constrained systems are situated in dynamic environments [9].

Computational intensive data processing often uses a big data cloud model, where the semantic data is uploaded in real-time to remote high resource servers for data processing and storage over high capacity links, but such an approach faces several as yet unsolved challenges [10], [11]. In terms of the use of semantic computing for quick onset EWS applications, disruptions to the physical environments can disrupt the communication infrastructure leading to low or variable bandwidth availability. Big data processing tends to be designed for low priority batch-mode processing, rather than for high priority, time critical processing, e.g., for DSS. In addition, big processing is strongly oriented towards parallelising numerical computation so that this can complete more quickly, rather than on supporting high performance semantic data processing. Hence, our time-critical semantic computing EWS is designed to deal with a variable bandwidth network, with failed links, and to use a hybrid semantic data model and processing, leveraging the use of lightweight ontologies as much as possible.

Use of semantics to enhance (the upstream) data exchange at or near the environment sensor data sources may not be required as these tend to be designed to transmit data to a local sensor access node using relatively simple, proprietary, data structures and encodings. This multiplexed data from

multiple sensors and routes these to a remote data processing centre. Thus, sensors only need to simply interoperate with a control centre via a sensor's access node. However, multiple sensors' data may need to interoperate and be fused to enhance data processing. These data processes occur more downstream: semantic representations can be better added where the data is stored, not where it is generated. Only a few of the current proposed EWS designs tend to use a lightweight semantic design: e.g., UrbanFlood [12], DEWS [2] and [13]. Even fewer EWSs state that they use heavyweight semantic support but they give too few details to understand how and why such semantic models are specifically being used, e.g., SLEWS [14]. The development of shared domain-specific rich ontologies is challenging [15]. It often relies heavily on domain experts. Meta-data model driven approaches can reduce the reliance on the use of domain experts to validate operational semantic data model changes [16].

In terms of non-EWS type environment monitoring systems, first, semantics can be used to define a richer meaning for sensor data e.g., the W3C Semantic Sensor Network, SSN, [17] ontology. SSN adds lightweight semantics to concepts defined using the OCG's (Open GIS Consortium's) SWE (Sensor-Web Enablement) standard specifications. The main SSN ontology classes have been aligned with classes in the DUL (DnS Ultra Lite) foundational ontology, to facilitate reuse, interoperability and ontology alignment and matching [17]. However, each application tends to define their own different ontological commitments to use an ontology, and their own instantiations and extensions to it. For example, the SSN ontology can be used to promote automatic plug and play for sensors while the OCG SWE specifications cannot [18]. However, the SSN ontology does not specify types of observed properties but introduces a generic property concept for further sub-classing. Hence specific properties and feature types can be imported from other ontologies, e.g., the Semantic Web for Earth and Environmental Terminology (SWEET) [19]. Non-SSN, sensor data, ontologies and SPARQL, the SPARQL and RDF Query Language, can be used to query the ontology model but in some cases the justification for using the semantic model and its deployment details are weak [20].

The sensor context, such as space and time, can be represented in a richer semantic form, to better support conditional queries and to adapt data services to these contexts. Spatial and temporal extensions to RDF, stRDF, have been proposed, to develop a Semantic Sensor Web registry that can be queried in space and time [21]. The spatial-temporal context of citizens can also be used to alert targeted individuals [22]. Semantics can be used to enhance data processing such as fusion from multiple data sources and enhance queries and to adapt the results to support different ontological commitments [23]. Due to additional unexpected events – e.g. aftershocks – workflow plans may vary over time: other regions may become affected and different recommendations may have to be given. Semantics can be used to improve service discovery [24] and to enable more flexible, dynamic, work-flow or plans for services [25]. Services can be represented using semantic descriptors and different tech-

niques, such as automated planning [26], can then be applied. To conclude,

- Majority of current reported EWSs tend to use non-semantic models.
- Relatively few EWSs use lightweight semantic models, even less use heavyweight ones.
- Current reported EWSs do not take explicit account of practical system constraints such as being used in time-critical, high-demand and resource-constrained situations (to meet objective 1, see Section I.B).

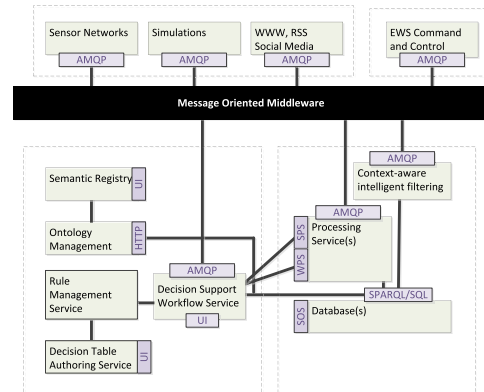


FIGURE 1. Overview of the semantic high-level IoT EWS architecture. Risk assessment is performed interactively by experts using the command and control UI. Assessments are based on visualizing raw heterogeneous information feeds, simulation results and analytic reports generated by decision support workflows and processing services.

III. SEMANTIC IOT EWS DESIGN AND IMPLEMENTATION

An overview of the high-level semantic IoT EWS architecture is given in Fig. 1. The overall data flow is that application specific (upstream) data flows are driven by fixed sensor data acquisition. Downstream, the main data flows are driven by the need to use the data for data fusion and mining, decision support and command-control driven workflows. Note that the semantic EWS system architecture offers generic semantic data analysis support. Hence, the domain-specific risk analysis is done at the application layer outside the system architecture. The design and implementation of the main components of the semantic EWS are given in the following sections. The main components are as follows: a Message-Oriented Middleware (MOM) service is used both to manage the lightweight semantic message exchange upstream to the data store, and to support the heavyweight semantic message exchange for downstream Data Fusion, the Decision Support System (DSS) and for workflow services.

A. MESSAGE-ORIENTED MIDDLEWARE (MOM)

A federated MOM system is used to manage the data exchange with lightweight semantics across the whole distributed semantic EWS as a system-of-systems. There are two benefits in using a MOM:

- It supports asynchronous data exchange between multiple publishers (data sources or sinks) and multiple

consumers (*data services*) as well as synchronous data exchange.

- *It decouples these from each other via a message broker so that new ones can be added and old ones can be removed, more flexibly at runtime.* This decoupling enables sensor data to be published at a faster rate using lightweight semantic mark-up, i.e., using the MOM topic namespace model.

Heavyweight semantics can be added and linked via additional metadata when the sensor data is imported in the knowledgebase (Section III.B). MOMs support highly scalable message exchange, e.g., a multi-core MOM server can handle throughputs of up to the order of 100 million messages per second over a fast dedicated LAN. However, in practice, the throughput is far more limited due to the propagation delay caused by physical environment changes that disrupt the communication bandwidth availability of the local access loop, especially when using a shared public WAN or LAN rather than using a dedicated end-to-end network. A MOM supports basic resilience for the message broker via simple mirroring and guaranteed message delivery.

The MOM is implemented as an extension of Apache Qpid that supports the use of a standard binary encoded message exchange protocol AMQP (Advanced Message Queuing Protocol) to enhance interoperability rather than supporting a (programming language) specific message API. *First, the extended MOM improves the basic resilience of the standard message broker to prevent it becoming overloaded, i.e., by rogue publishers flooding the broker with large fake messages, by high-rate messaging, and by publishing unneeded topic messages. Second, the extended MOM prevents rogue slow rate subscribers causing messages to build up in the broker [27].* Brokers can be organized into one or more interlinked broker clusters with each cluster organised as a hierarchy of a head broker and two or more edge ones, to aid scalability and resilience (see Section IV.A). The extended Qpid MOM does not instrument or modify the broker itself to enable this enhanced scalability and resilience, but uses a special client of the broker, called a Management Agent (MA) that interfaces it via a system management API such as the Java Management eXtension or JMX. Broker management agents use a subset of AMQP to exchange information about the load of any attached publishers and subscribers with each other. The MAs can be used to achieve a Load Balancing Head Edge Broker Overlay or LBHERO for brokers [27]. The broker load metrics are described in Section IV.A.

The upstream sensor (message publisher) data exchange to the broker is not designed to support heavyweight semantics. Such semantics is added downstream. The upstream message broker itself does however support lightweight semantics, i.e., topic (name) matching [27]. Two example topic subscriptions using a wildcard “*” are given below:

```
“Bodrum.EastMediterranean.SeaLevel.SeaServiceHeight.*”.  
“Bodrum.EastMediterranean.SeaLevel.*.*”
```

The 1st one is used to subscribe to any measurements produced by the SeaServiceHeight sensor. The 2nd one subscribes to only SeaLevel measurements, regardless of the sensor used. The topic namespace and its hierarchical data

structure are mapped to the application domain specific part of the ontology model used by the semantic registry (Section III.C). Because the upstream sensor data exchange needs only to support very simple workflows for data to reach the sensor data repository, and because new types of fixed sensor are seldom added to the operational system, the need for heavyweight SSN ontology to support plug and play is not required for the upstream exchange in our semantic EWS.

B. KNOWLEDGEBASE, DATA FUSION AND MINING SERVICES

The Semantic EWS Knowledgebase (KB) is much more than a basic database, it holds a wide variety of data at different semantic levels. A real-time database feeder filters and caches sensor data in a scalable way, transcoding MOM messages using a variety of domain semantics and making them available as a common database layer in the KB. Raw sensor upstream measurement data is stored using the Open Geospatial Consortium (OGC, see <http://www.opengeospatial.org>) Observation and Measurement (O&M) model, which defines measurement concepts, units, allowed values and uncertainty information. *Data and metadata are deliberately stored separately, allowing faster, more efficient SQL/NoSQL lookups on large amounts of raw data versus slower but more expressive SPARQL queries on the metadata.*

The KB holds the result sets that are continually generated and updated by online data-mining and data-fusion techniques, each producing data at a variety of semantic levels. Some data describes the features and patterns discovered in a domain. Other data represents reports from domain experts and other data represents the knowledge extracted by off-line semi-manual data-mining and data-fusion techniques. The stored data elements are mapped to the decision support upper ontology (Section III.C), to ensure that the concepts are semantically grounded in a common understanding.

In more detail, *the semantic data fusion services are responsible for combining and analysing data or information from different sources to estimate or predict the states of entities existing in the problem domain or the occurrence of events of interest.* The ‘knowledge-base’ uses a variety of data fusion algorithms and models wrapped as OGC remote Web Processing Service (WPS) or OGC Sensor Planning Service (SPS). Multiple levels of data are stored, based upon use of the Joint Directors of Laboratories (JDL) data fusion model [28]. These levels are:

- *Level 0 (Pre-Processing):* this allocates data to appropriate processes. It selects appropriate sources and data adjustments to attain a common data structure. It uses noise reduction and deals with missing data.
- *Level 1 (Object Assessment):* transforms data into a consistent structure for discovery of features and patterns, data and object correlation, hypothesis formulation and feature extraction.
- *Level 2 (Situation Assessment):* provides a contextual description of relationships among objects and observed events, using a-priori knowledge and context information and models errors and uncertainty.

- *Level 3 (Impact Assessment)*: evaluates the current situation, projecting it into the future to identify forecasts and inferring possible impact based on multi-perspective assessments. This level includes the data processing required for decision support.
- *Level 4 (Process Refinement)*: is considered outside the domain of our specific data fusion functions.

Note that the *SSN ontology type services surveyed (in section II) focus on support for data fusion levels 0-1 only. We support more data fusion levels, 0-3. In our Semantic EWS, result sets are explicitly stored at different fusion levels as separate database entries. This aids decoupling algorithms from the data, encouraging agile composition of processing services working at different semantic levels and provides decision support actors with the ability to drill down and review data at different semantic levels, helping them to fully understand the context in which knowledgebase results are presented.*

The access to the data-fusion functionality is achieved via the OGC WPS and SPS services. The resulting data is accessible as a result of an OGC Sensor Observation Service (SOS) call or directly via SPARQL/SQL queries to the result databases. WPS processes and SPS tasks can be configured, and re-configured, to factor in contextual information available at any moment in time. Algorithms run continuously over long periods of time to receive and process raw data updates, checking the databases via polling SPARQL/SQL queries or receiving event streams directly via defined APIs. Real-time updates to their configuration via contextual steering, driven from the intelligent context processing are also supported. A process steering component sets up and manages processing pipelines of WPS and SPS services, each providing access to specific algorithms and models.

C. SEMANTIC REGISTRY, DECISION SUPPORT ONTOLOGIES (DSO) AND SERVICES

The Semantic Registry or repository offers the ability to publish, search, query and retrieve descriptive information (meta-information) for resources (i.e. data and services) of any type, in a standardized manner, across the whole EWS distributed system. Its ontology data model links all other services and their data together. The Ontology Store part of the semantic registry is used to store and maintain the DSO (see Fig. 2).

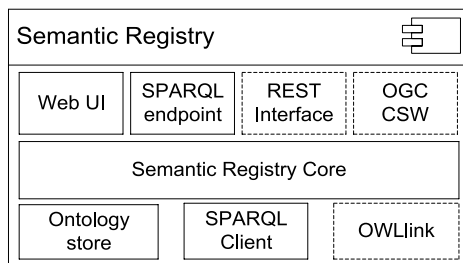


FIGURE 2. Components of the Semantic Registry.

There are several interfaces to the Ontology Store:

- A SPARQL endpoint and client act as a proxy to the triple-store that backs on to the Semantic Registry.

- A RESTful service interface maps REST (Representational state transfer) operations to semantic queries, allowing client applications to execute complex queries without requiring support of semantic web standards and SPARQL.
- A Web-based User Interface and further interfaces, e.g. an OGC conformant Catalogue Service and OWLlink (see <http://www.owllink.org/>), can be adapted for use with specific applications.

The main challenge in the design of the DSO is to adequately adapt the concepts to the objects (e.g. sensors, data streams) and operational procedures, which govern the management of a crisis. The design of the DSO is based on a top down approach by re-using and extending ontological patterns from available ontology sources, and a bottom-up approach by designing thematic models derived from use-cases found in the domains for the NCM and ISD scenarios. The top-down development of the DSO involves a collaborative effort amongst domain experts and data contributors. As these are generally not experts in ontology engineering, we set up a development process that only required a minimum of expertise about the principal ontological elements. An agreement on a common terminology had to be reached which mediates between domain experts (who have the knowledge about NCM and ISD domains, who possibly speak different languages and who may have distinct responsibilities and play different roles) and IT experts (who have the knowledge about specific technological vocabularies, but might lack the necessary domain knowledge for deciding on the right course of action as the crisis evolves).

The need to extend a standard ontology to support different applications' Ontological commitments has already been mentioned (Section II). The design of the *Decision Support Ontology (DSO)* supports four requirements: to express sensor measurements with a spatial context, their measurement units, their time context and the event context. The DSO uses the W3C SSN ontology [17] as a base ontology, to express the sensor measurements with a spatial context. This is aligned to the OGC sensor device standards, e.g., WPS, SPS, and SOS but while these OGC standards provide description and access to data and metadata for sensors, they do not provide facilities for abstraction, categorization, and reasoning that are offered by semantic technologies. Hence, the *DSO* is designed to aggregate and align multiple ontologies to support compound EWS semantics and ontology commitments as follows:

- *SSN ontology does not define a system of units and quantities* to enable measurements in different units to be combined. Hence, a Measurement Units (MU) ontology represented in OWL [29] is added and aligned with concepts in SSN as part of the DSO.
- *SSN inherently supports spatial properties but it does not define support for temporal concepts.* The OWL-Time ontology [30] is used to capture topological relations among instants and intervals, together with information about durations, and about date-time information, and integrated into DSO.

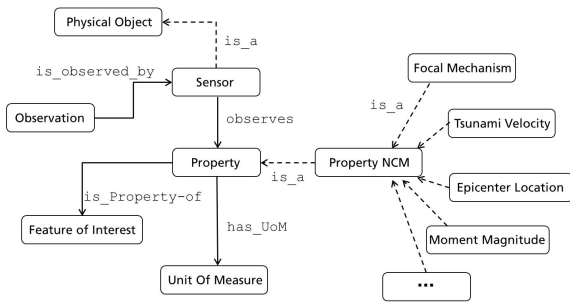


FIGURE 3. Excerpt of Concepts contained in the DSO.

- DSO integrates concepts set of ontologies from SWEET for the geo-science domain [19].
- DSO also integrates an event ontology to express any events detected in real time [31].

These events arise from complex correlations of measurements made by independent sensing devices. Because the mapping of such complex events to direct sensor measurements may be poorly understood, such methods must also support experimental and frequent re-specification of the events of interest. This means that the event specification method must be embedded in the problem domain of the end-user, must support user discovery of the observable properties of interest, and must provide automatic and efficient enacting of the specification.

The example in Fig. 3 illustrates an excerpt of DSO showing the main relationships of SSNO (Semantic Sensor Network Ontology) concepts “Sensor” and “Property”. Sensors defined as (DUL) “Physical Objects” attached to a SSNO “Sensing Device”. Properties are qualities that can be observed by a certain kind of sensor; they infer the SSNO Features of Interest, which are entities in the real world that are the target of sensing. A Property has relationships to classes defined by the upper ontologies (e.g. Unit of Measure) and to subclasses which have been defined for the TRIDEC domains (e.g. “Tsunami Velocity” or “Focal Mechanism” defined in for the NCM domain).

Although SSN was extended to be combined with MU, OWL-TIME, SWEET and Event ontologies to form the DSO, domain specific ontology adaptation is still needed. Initially, the SSN ontology formed the main conceptual backbone of our approach, however, these remain very high level specifications offering very generic terms and attributes. In contrast, the terminological definitions found in specific application domains are very concrete and focused. Moreover, in the ISD domain for instance, properties typically have different names depending on the users’ roles and views, i.e. we often found many definitions for identical items. Thus, we had to provide the means to identify the different items and to additionally find adequate mappings to the definitions given in DSO. This involved not only a great deal of work for the ontology mappings at a technological level, but also involved many discussions with domain experts in order to find the correct mappings and to use the available ontologies properly [33]. The DSO is formally represented in OWL, containing description logic (DL) expressions. These are hard to

understand by, and somewhat too generic for, non IT-experts, hence, this process needs much mediation and guidance by the experts who developed the formal ontology.

When filling up the Semantic Registry with descriptions of concrete objects (e.g. sensors, properties) data entered follows the ontological concepts defined for these objects. For instance, the data entered for a sensor comprises specific relations of this sensor, e.g. the properties it observes and the system to which it is attached. The forms for entering these definitions are generated automatically from the SSN ontology definitions. As mentioned above, these descriptions are quite exhaustive and comprise many attributes and relations. Consequently, the generated forms comprise a large number of entered data. Most of this data is not needed in our application context, but the forms appear large and awkward to the user.

Hence, in order not to deter users from giving inputs, we developed a solution with slim forms which fits the input to the needs of the application as follows:

- We used a selection to fit the application-driven ontology requirements, not the complete SSN and DUL ontologies.
- We developed a mechanism by which the administrator of the Semantic Registry can easily select those relations of concepts which should appear in the input forms.

D. WORKFLOW SERVICE AND RULE ENGINE

Current operational EWS systems tend to use hard-coded information logistics processes even though they are subject to change. In addition, systems are tailored to the policies and requirements of a certain organization and changes can require major refactoring. Hence, our workflow management system (WfMS) was designed to meet these requirements:

- It can be deployed and adapted to multiple organizations with different policies.
- Changes can be applied locally, without affecting the larger parts of the system.
- Extensibility: new services and information sources can be integrated and used within DSS workflows.

As business processes and emergency plans are similar, the use of WfMS for automating and managing emergency plans has been proposed [32]. Hence, a standard solution is adopted to use WfMS that execute workflows modelled using graphical notations, such as BPMN2 (Business Process Model and Notation 2.0, see <http://www.bpmn.org/>). Note that workflow models are used more to govern the more complex downstream information dissemination in the system to the stakeholders rather than to govern the simpler continuous upstream operational data processes for data acquisition, knowledgebase updates.

At the core of the Workflow Service is Activiti (<http://activiti.org/>), an open-source BPMN2 workflow engine that in addition manages workflow deployments and monitors and tracks the history of workflows. The Workflow Service is accessed via a web-based user interface and a RESTful HTTP interface (Fig. 4). Workflows can be authored offline using a BPMN2 editor and then deployed via a RESTful interface.

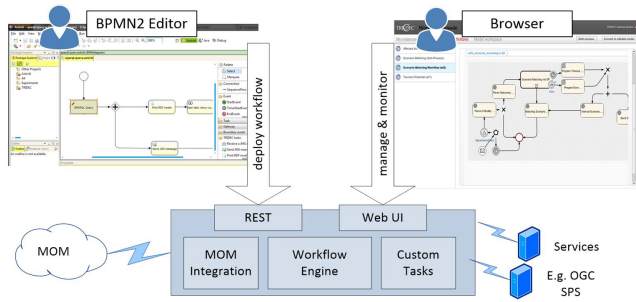


FIGURE 4. Interfaces of the Workflow Service.

The Workflow Service integrates the workflow engine with the MOM via additions to the workflow engine that parse each new deployed workflow in order to update the necessary MOM topic subscriptions, which enable workflows to interact with existing and newly developed services. This enables any MOM topic to be used within message and signal events and hence within workflows. All MOM subscriptions are handled dynamically.

Workflows often include rules that determine, for example, under which circumstances certain services are invoked or alert messages are sent. These rules can in principle be encoded in BPMN2 using branches and conditions. However, *rules are separated from workflows for two main reasons.*

- When rules become complex, the resulting workflow becomes difficult to understand and to maintain.
- If rules change separately from the general workflow, different versions of rule sets can be tested without modifying the overall workflows.

This separation can reduce the complexity for users at the user interface to allow changing rules without dealing with the possible complexity of workflows. While various representations for rules exist, an empirical evaluation of the comprehensibility of decision tables, decision trees and textual propositional rules showed that decision tables perform significantly better against other formats under consideration (binary decision trees, propositional rules and oblique rules) on all three criteria applied in an end-user experiment (accuracy, response time and answer confidence for a set problem-solving tasks involving the above representations) [34]. Additionally, a majority of the users found decision tables the easiest representation format to work with. These findings corresponded with our experience that decision tables can be used for communicating rules. Consequently, decision tables are integrated into the decision support and workflow system.

The Drools Expert rule-engine (<http://www.drools.org/>) is used to evaluate rule sets. However, the rule sets represented as decision tables are not edited directly but instead edited using a custom editor or using a spread sheet application. Decision tables are then compiled into rule sets which can be used within workflows.

IV. VALIDATION AND RESULTS

Two main types of validation are undertaken for the Semantic IoT EWS system:

- *Non-functional (scalability and resilience) tests* were performed for the upstream system components that needed to be scalable, for the MOM, and for the knowledgebase. The downstream system interaction for DSS and workflows is more complex and application specific, and its throughput performance is far lower than the upstream message exchange performance.
- *Functional validation* of the semantic EWS was performed in two different application domains: tsunami NCM and ISD but here the focus is on the tsunami NCM. These were done as part of the EU FP7 funded TRIDEC Collaborative, Complex and Critical Decision-Support in Evolving Crises) project.

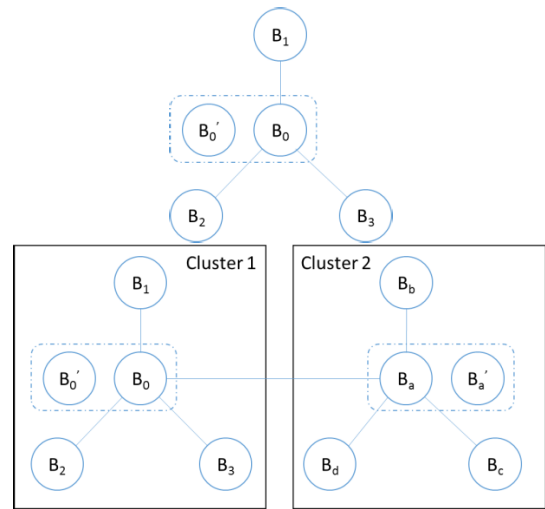


FIGURE 5. Broker deployment as a single cluster deployment (top) versus federated cluster deployment (bottom).

A. NON-FUNCTIONAL TESTS (SCALABILITY AND RESILIENCE)

These tests are divided into two:

1. Tests for the upstream lightweight sensor data and meta-data acquisition and exchange
2. Tests for the sensor data and semantic data annotation and storage to enable downstream heavyweight semantic data driven processing [36].
3. We tested MOM performance, in terms of scalability and resilience in order to exchange data and metadata in both multi-broker, single cluster and multi-broker, multi-cluster settings, (see Fig. 5). In our experimental testbed, message brokers run in different virtual machines (VMs) on the same server or on different servers (typically, with a 2.3 GHz CPU, 4 GB memory and 100 Mbps bandwidth). The single cluster deployment (Fig. 5, top) consists of one head broker cluster (active head broker B_0 and backup broker $B_{0'}$) connected to three edge brokers (B_1 , B_2 , and B_3). It forms a star structure that mimics a cluster at a typical data centre as found in practice [35]. The federate deployment consists of two clusters that are connected via two head brokers (B_0 and $B_{a'}$) as shown in Fig. 5, bottom. In each cluster of the head-edge model, message consumers or

subscribers only connect to edge brokers; while message producers connect to edge brokers if there are only local subscribers, i.e., subscribers in the same cluster that subscribe on the same topic. If there are remote subscribers, i.e., subscribers in a neighbour cluster that subscribe to the same topic, publishers publish messages to the cluster-head broker.

Providing the broker runs on a high capacity server, it is well able to cope with the message rate load. However, on a lower capacity server its load may be exceeded. In a MOM broker the load in the broker is measured in terms of the message queue which increases when publishers publish on a topic to a broker versus decreases when a subscriber subscribes to a topic in a broker. A queue builds up when the message input rate from a publisher exceeds the message output (or consumption) rate by a subscriber for that topic. Experiments to test how a federated broker handles a potential broker overload and triggers load rebalancing are given as follows.

Each experiment is divided into three phases: 1) client distribution phase: 1s – 15s, subscribers of each topic in a EWS are registered and distributed to the available brokers in each second; 2) equilibrium phase: 15s - 29s, both publishers and subscribers in a EWS run without message bursts or client joining or leaving; 3) message burst simulation and offloading phase: at 30s, a burst that simulates a message flood when a crisis detected is generated by doubling the speed of publishing 7 topics (e.g., topic 2, 4, 6,..., 12, 14); after 31s, up to the end of the experiment, offloading will be triggered if any load metric exceeds its higher threshold, i.e., a broker becomes overloaded.

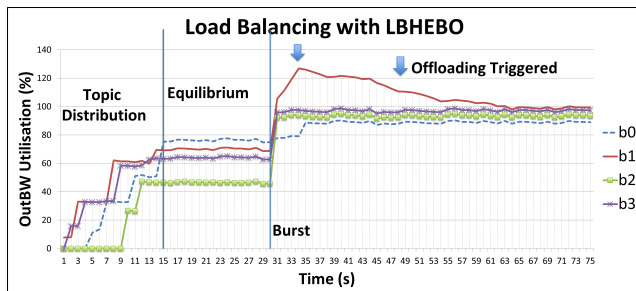


FIGURE 6. Simulation Result for OutBW Utilisation with LBHEBO.

The duration of each phase does not affect the behaviour of the system. The main reason to set the time slots to these values is to highlight the changes in each stage of the simulation. Fig. 6 shows the simulation results for the outBW utilisation in percent (y) against time in second (x) in one experiment. After the load distribution, broker b1 serves topics 1, 3, 8, and 14 (see to the 4 inflection points of b1 in the topic distribution stage, Fig. 6). For b1, the output queue starts to build up after a message burst at 30s as the outBW utilisation exceeds 100%; 4s after the burst (34s), the queue depth value of topic 8 exceeds TH_{high} , and thus offloading is triggered. Topic 1 in b1 is migrated to broker b0. Therefore, a broker b1 has more bandwidth to clear the messages for topic 8 in the queue (from 34s – 62s, a balancing stage). After 62s,

the message queue for topic 8 in broker b1 is removed. The outBW utilisations for all the brokers are below 100%.

In addition, a detailed evaluation of the knowledge-base storage and retrieval performance was performed through comparing different database approaches to store semantic data structures in the form of triples that included 4-store (<http://4store.org/>), OWLIM (now called GraphDB, www.ontotext.com/products/ontotext-graphdb/), MySQL (<http://www.mysql.com/>) that were combined with the prototype database feeder module. Of these, 4-store does not support multi-client connections for data importing (a serious flaw) hence we discounted it. In an experiment we ran many clients, each importing data into the database in parallel (see Fig. 7) to see how each databases performance is effected by multiple clients populating it with data in parallel.

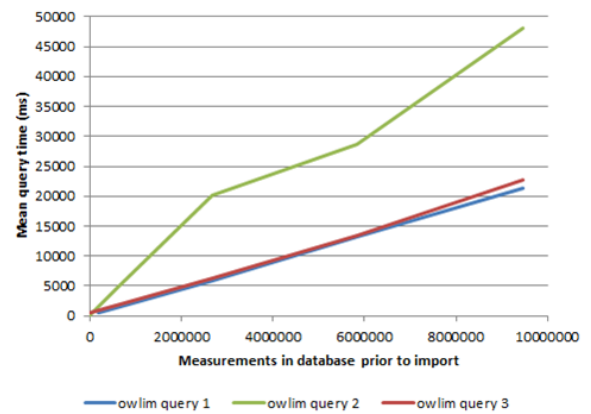


FIGURE 7. Query speed as a factor of OWLIM-lite database volume.

This test gives us an insight into how many data sources and database feeders are practical to use with each database solution. An alternative to store and retrieve semantic data structures is to use non-relational databases (i.e., NoSql solutions). Most of the data storage technologies used for Big Data fall into this category such as Google’s BigTable, Amazon’s Dynamo and open source databases such as Apache’s Cassandra and MongoDB (see <http://www.mongodb.org>).

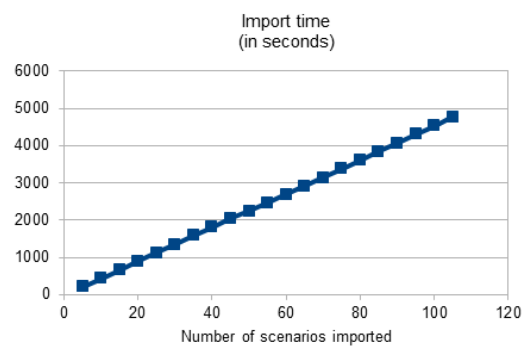


FIGURE 8. Simulation import time in MongoDB.

An example use of a NoSql approach is as part of a tsunami scenario matching service that allows users to retrieve a set of tsunami simulations that have been previously pre-computed and stored in the system. The retrieval task is driven by a

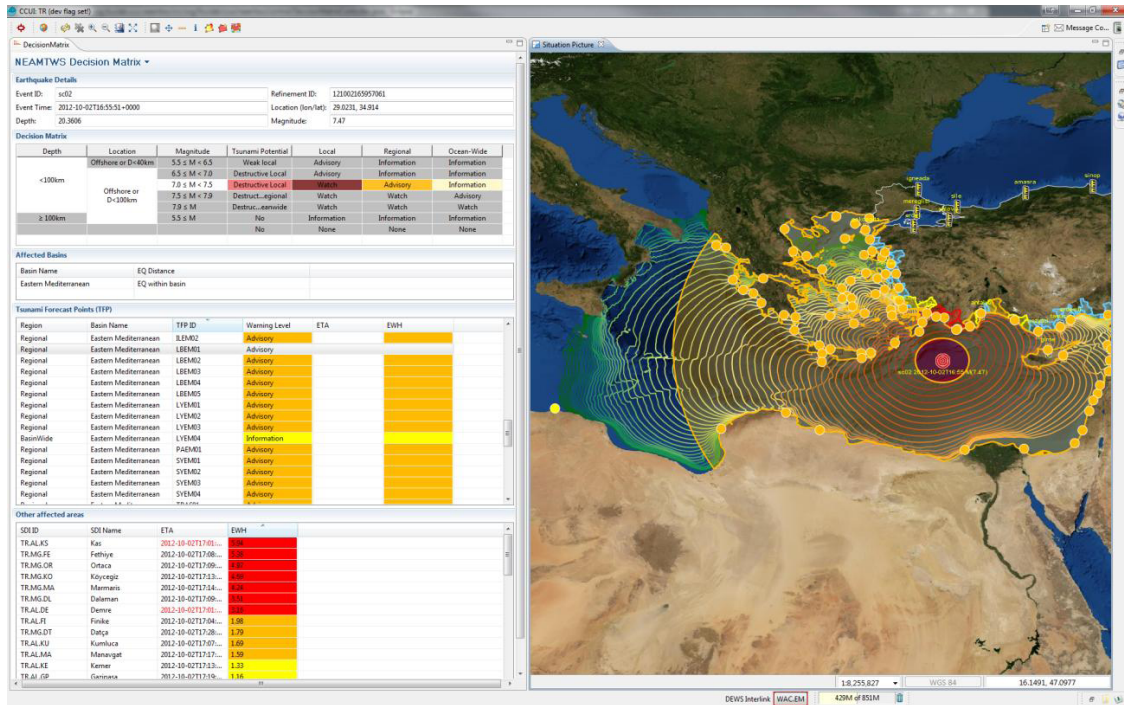


FIGURE 9. Screenshot from TRIDECC command and control user interface (CCUI) taken during NEAM Wave 12 exercise. The contours represent a spatial-temporal view of a tsunami simulation for the exercise's earthquake event in the Eastern Mediterranean. The coloured circles represent anticipated tsunami impact at pre-determined tsunami Forecast Points on the coast where warnings should be disseminated to the general public through civil protection authorities via channels such as SMS, Email & Twitter.

concept of similarity between the recorded event and the simulated one which is twofold. A tsunami can be compared either by: seismic parameter similarity, or by water height distribution similarity. The first similarity concept requires the similarity to be computed over the recorded parameters that are stored. Once a set of similar scenarios have been identified the system can extract the simulation data and the measure of similarity of the water height distributions. The similarity is computed by comparing the water height distributions. The computation performance is mainly influenced by the size of the data cubes which are stored and retrieved as binary blobs by the service. For testing the behaviour when importing the simulations, we first used a typical data cube from a typical scenario (1.3Gb in size each) and recorded the import time at different stages. The resulting distribution shows that despite the time to import a single scenario being around 45 seconds, it remains constant even when the number of scenarios stored in the system increases (see Fig. 8).

B. FUNCTIONAL TESTS (TSUNAMI NCM)

On November 27-28, 2012, the Kandilli Observatory and Earthquake Research Institute (KOERI) joined other countries in the North-Eastern Atlantic, the Mediterranean and connected seas (NEAM) region as participants in an international tsunami response exercise. The exercise, titled NEAMWave12, simulated widespread tsunami watch situations throughout the NEAM region. It was the first international exercise in this region where the UNESCO-IOC

ICG/NEAMTWS (Intergovernmental Coordination Group for the NEAM region Tsunami Warning System) had been tested, full scale, with different systems, including the semantic EWS which was developed as part of the TRIDECC project [37], see Fig. 9.

Because tsunami occurrences in specific regions tend to be relatively infrequent, tsunami EWS system tests typically involve the use of simulated tsunami data and events, e.g., using the SeisComP seismological software simulator (<http://www.seiscomp3.org/>) to support data acquisition, processing, distribution and interactive analysis, the MOD1 (Model1) tsunami Scenario Database and TAT (Tsunami Analysis Tool) [38]. NEAMWave12 involved the simulation of the assessment of a tsunami, based on an earthquake-driven scenario followed by alert message dissemination by Candidate tsunami Watch Providers-CTWP (Phase A). It continued with the simulation of the tsunami Warning Focal Points/National tsunami Warning Centres (TWFP/NTWC) and Civil Protection Authorities (CPA) actions (Phase B), as soon as messages produced in Phase A have been received. Phase A covers the simulation of a tsunami assessment triggered by an earthquake scenario, tsunami alert message dissemination by CTWP and the message reception and evaluation by tsunami Warning Focal Points (TWFP). Each CTWP selected one single earthquake scenario and computed the corresponding prescheduled tsunami assessment. The exercise included the dissemination of 4 messages at the 10th, 25th, 62nd and 180th minutes of the scenario event, respectively. KOERI exploited the TRIDECC system

in addition to the existing operational infrastructure, especially making use of artificial eye-witness reports sent and geographically referenced by the Geohazard Android Application [39] and the open-source crowd-mapping platform Ushahidi (<http://www.ushahidi.com/>).

The tsunami scenario database used by KOERI is based upon code that solves the shallow water equations using a finite difference numerical scheme. Initial conditions for the tsunami model are obtained using an analytical solution for surface deformation in an elastic half-space by estimating the distribution of co-seismic uplift and subsidence using the earthquake source parameters. The code is validated by first initialising the calculation space and then performing the travel time propagation calculation. At each step the locations reached by the wave are verified and thus the visualization and animation files are updated [40]. *In addition to providing synthetic test sensor data measurements representing a tsunami occurrence*, there are two further uses of the tsunami simulations.

- *Simulations can be applied pre-emptively, to the decision support system* in order to assess a predicted tsunami as early as possible, before enough real observations from sea level sensors are available.
- *Reverse computing (predicting) the sensor observations (synthetic time series) from the simulated wave propagation can be used to verify a tsunami assessment* by matching synthetic data with real data (as soon as they are available) in order to confirm or take back the predictions made.

User-tailored warning messages with customization based on recipients' vocabulary, language, subscribed region, criticality, and channel have been generated and disseminated to the Turkish CPA via email and to other registered message recipients via FTP (imitating the Global Telecommunications System, GTS, network for the transmission of meteorological data), email and SMS as well as social media channels via installations of the twitter clone StatusNet and a WordPress blog. Exercise messages were disseminated containing hazard maps with the affected coastal zones possibly being exposed to the tsunami inundation as well as containing the same content as the NEAMTWS messages. Again, the direct centre-to-centre communication with the TRIDEC system deployed at IPMA (Instituto Portuguese do Mar e Atmosfera) was exercised [37].

V. CONCLUSIONS

Based upon our experiences of developing a semantic IoT EWS, the following emerging trends are identified in order to more effectively apply the use of semantic computing models for use with EWS type environments.

1. *In practice, heavyweight semantics should be selectively used in specific parts of a distributed, multi-sensor IoT* as the use of heavyweight semantics requires substantive computation and memory use that may not be available in low resource sensor things.
2. *Support for multiple levels of semantics and mapping between them are needed, i.e., between lightweight and heavyweight representations.*

3. *Multiple domain ontologies may need to be combined*, in part because of the cross-disciplinary concepts used by stake-holders of a domain specific IoT; multiple knowledge representations need support from a range of data fusion algorithms.
4. *Some higher-level abstractions and user interfaces to the semantic models are needed* for use by domain experts who are perhaps not experts in semantic modelling, to ease their input and their manipulation of these.
5. *The use of semantic computing models in specific application domain IoTs needs to be tempered in practice according to their operational constraints*, e.g., for EWSs these affect the time-critical, scalable, resource-constrained and resilient data (and metadata) exchange and management.

Although, we oriented our discussion of the application of semantics to IoT EWS use for natural crises management, IoTs for other application domains that share similar operational system constraints could also benefit from our design and implementation of a semantic computing system. These potential applications include financial and banking systems, health and physiological signal acquisition and monitoring, and smart transport and utility management in smart cities.

REFERENCES

- [1] A. Wiltshire, "Developing early warning systems: A checklist," in *Proc. 3rd Int. Conf. Early Warning (EWC)*, 2006.
- [2] D. Jin and J. Lin, "Managing tsunamis through early warning systems: A multidisciplinary approach," *Ocean Coastal Manage.*, vol. 54, no. 2, pp. 189–199, 2011.
- [3] R. D. Braddock, "Sensitivity analysis of the tsunami warning potential," *Rel. Eng. Syst. Safety*, vol. 79, no. 2, pp. 225–228, 2003.
- [4] W. Hanka et al., "Real-time earthquake monitoring for tsunami warning in the Indian Ocean and beyond," *Natural Hazards Earth Syst. Sci.*, vol. 10, pp. 2611–2622, 2010.
- [5] M. Dorasamy, M. Raman, and M. Kaliannan, "Knowledge management systems in support of disasters management: A two decade review," *Technol. Forecasting Soc. Change*, vol. 80, no. 9, pp. 1834–1853, 2013.
- [6] D. L. McGuinness, "Ontologies come of age," in *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*, D. Fensel, J. Hendler, H. Lieberman, and W. Wahlster, Eds. New York, NY, USA: MIT Press, 2003, pp. 171–196.
- [7] J. Davies, "Lightweight ontologies," in *Theory and Applications of Ontology: Computer Applications*, R. Poli, M. Healy, and A. Kameas, Eds. Amsterdam, The Netherlands: Springer-Verlag, 2010, pp. 197–229.
- [8] S. Poslad, "Intelligent systems (IS)," in *Ubiquitous Computing: Smart Devices, Environments and Interactions*. Chichester, U.K.: Wiley, 2009, pp. 263–268.
- [9] L. P. Kaelbling, "A situated-automata approach to the design of embedded agents," *ACM SIGART Bull.*, vol. 2, no. 4, pp. 85–88, 1991.
- [10] P. Barnaghi, A. Sheth, and C. Henson, "From data to actionable knowledge: Big data challenges in the Web of Things" *IEEE Intell. Syst.*, vol. 28, no. 6, pp. 6–11, Nov./Dec. 2013.
- [11] C. Bizer, P. Boncz, M. L. Brodie, and O. Erling, "The meaningful use of big data: Four perspectives—Four challenges," *ACM SIGMOD Rec.*, vol. 40, no. 4, pp. 56–60, 2011.
- [12] B. Balis et al., "The urbanflood common information space for early warning systems," in *Proc. Int. Conf. Comput. Sci. (ICCS)*, vol. 4, 2011, pp. 96–105.
- [13] M. Lendholt and M. Hammitzsch, "Generic information logistics for early warning systems," in *Proc. 8th Int. Conf. Inf. Syst. Crisis Response Manage.*, 2011.
- [14] C. Arnhardt et al., "Sensor based Landslide Early Warning System-SLEWS. Development of a geoservice infrastructure as basis for early warning systems for landslides by integration of real-time sensors," *Geotechnologien science report*, vol. 10, pp. 75–88, 2007.

- [15] F. Fiedrich and P. Burghardt, "Agent-based systems for disaster management," *Commun. ACM*, vol. 50, no. 3, pp. 41–42, 2007.
- [16] S. H. Othman and G. Beydoun, "Model-driven disaster management," *Inf. Manage.*, vol. 50, no. 5, pp. 218–228, 2013.
- [17] M. Compton et al., "The SSN ontology of the W3C semantic sensor network incubator group," *Web Semantics, Sci., Services Agents World Wide Web*, vol. 17, pp. 25–32, Dec. 2012.
- [18] A. Bröring, P. Maué, K. Janowicz, D. Nüst, and C. Malewski, "Semantically-enabled sensor plug & play for the sensor Web," *Sensors*, vol. 11, no. 8, pp. 7568–7605, 2011.
- [19] R. G. Raskin and M. J. Pan, "Knowledge representation in the semantic Web for Earth and environmental terminology (SWEET)," *Comput. Geosci.*, vol. 31, no. 9, pp. 1119–1125, 2005.
- [20] K. Ramar and T. T. Mirmalinee, "An ontological representation for tsunami early warning system," in *Proc. Int. Conf. Adv. Eng., Sci. Manage. (ICAESM)*, 2012, pp. 93–98.
- [21] K. Kyzirakos, M. Karpathiotakis, and M. Koubarakis, "Developing registries for the semantic sensor Web using stRDF and stSPARQL," in *Proc. 3rd Int. Workshop Semantic Sensor Netw. (SSN)*, 2010.
- [22] U. Meissen and A. Voisard, "Increasing the effectiveness of early warning via context-aware alerting," in *Proc. 5th Int. Conf. Inf. Syst. Crisis Response Manage. (ISCRAM)*, 2008, pp. 411–440.
- [23] S. Poslad and L. Zuo, "An adaptive semantic framework to support multiple user viewpoints over multiple databases," in *Advances in Semantic Media Adaptation and Personalization (Studies in Computational Intelligence)*, vol. 93. Berlin, Germany: Springer-Verlag, 2008, pp. 261–284.
- [24] C. Huang, G. M. Lee, and N. Crespi, "A semantic enhanced service exposure model for a converged service environment," *IEEE Commun. Mag.*, vol. 50, no. 3, pp. 32–40, Mar. 2012.
- [25] A. Ordonez, V. Alcázar, J. C. Corrales, and P. Falcarin, "Automated context aware composition of advanced telecom services for environmental early warnings," *Expert Syst. Appl.*, vol. 41, no. 13, pp. 5907–5916, 2014.
- [26] S.-C. Oh, D. Lee, and S. R. T. Kumara, "A comparative illustration of AI planning-based Web services composition," *ACM SIGecom Exchanges*, vol. 5, no. 5, pp. 1–10, 2006.
- [27] R. Tao, S. Poslad, and J. Bigham, "Resilient delay sensitive load management in environment crisis messaging systems," in *Proc. Int. Conf. Syst. Netw. Commun. (ICSNC)*, 2013, pp. 6–11.
- [28] D. A. Lambert, "A blueprint for higher-level fusion systems," *Inf. Fusion*, vol. 10, no. 1, pp. 6–24, 2009.
- [29] H. Rijgersberg, M. van Assem, and J. Top, "Ontology of units of measure and related concepts," *Semantic Web*, vol. 4, no. 1, pp. 3–13, 2013.
- [30] F. Pan and J. R. Hobbs, "Temporal aggregates in OWL-time," in *Proc. 18th Florida Artif. Intell. Conf. (FLAIRS)*, 2005, pp. 560–565.
- [31] K. Taylor and L. Leidinger, *Ontology-Driven Complex Event Processing in Heterogeneous Sensor Networks (Lecture Notes in Computer Science)*, vol. 6644. Berlin, Germany: Springer-Verlag, 2011, pp. 285–299.
- [32] C. Sell and I. Braun, "Using a workflow management system to manage emergency plans," in *Proc. 6th Int. ISCRAM Conf.*, 2009.
- [33] F. Chaves, J. Mossgraber, M. Schenk, and U. Bugel, "Semantic registries for heterogeneous sensor networks: Bridging the semantic gap for collaborative crises management," in *Proc. 24rd Int. Conf. Database Expert Syst. Appl. (DEXA)*, Aug. 2013, pp. 118–122.
- [34] J. Huysmans, K. Dejaeger, C. Mues, J. Vanthienen, and B. Baesens, "An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models," *Decision Support Syst.*, vol. 51, no. 1, pp. 141–154, 2011.
- [35] A. K. Y. Cheung and H.-A. Jacobsen, "Load balancing content-based publish/subscribe systems," *ACM Trans. Comput. Syst.*, vol. 28, no. 4, 2010, Art. ID 9.
- [36] S. E. Middleton, Z. A. Sabeur, P. Löwe, M. Hammitzsch, S. Tavakoli, and S. Poslad, "Multi-disciplinary approaches to intelligently sharing large-volumes of real-time sensor data during natural disasters," *Data Sci. J.*, vol. 12, pp. 109–113, 2013.
- [37] M. Hammitzsch et al., "Meeting UNESCO-IOC ICG/NEAMTWS requirements and beyond with TRIDEC's crisis management demonstrator for tsunamis," in *Proc. 23rd Int. Offshore Polar Eng.*, Anchorage, AK, USA, 2013.
- [38] N. M. Ozel et al., "Tsunami early warning in the Eastern Mediterranean, Aegean and Black Sea," in *Proc. 22nd Int. Offshore Polar Eng. Conf. (ISOPE)*, 2012, pp. 95–100.

- [39] M. Hammitzsch and M. Schroeder, "The Android app geohazard—Experiences with shared information on natural hazards," in *Proc. Joint W3C OGC Workshop On Linking Geospatial Data*, 2014. [Online]. Available: http://www.w3.org/2014/03/lgd/papers/lgd14_submission_7
- [40] N. Zamora, G. Franchello, and A. Anunziato, "Validation of the JRC tsunami propagation and inundation codes," *Sci. Tsunami Hazards*, vol. 33, no. 2, pp. 112–132, 2014.



STEFAN POSLAD (M'02) received the Ph.D. degree from Newcastle University. He is currently a Senior Lecturer/Associate Professor with the Queen Mary University of London. His research interests are Internet of Things, ubiquitous computing, semantic Web, and distributed system management.



STUART E. MIDDLETON received the Ph.D. degree in computer science from the University of Southampton (UoS). He is currently a Senior Research Engineer with the UoS IT Innovation Centre. His main research interests are social media, sensor systems, data fusion, and ontologies. He is a Senior Member of the Association for Computing Machinery.



FERNANDO CHAVES received the degree in computer science from the University of Karlsruhe, Germany. He is currently a Senior Scientist with the Fraunhofer Institute of Optronics, System Technologies and Image Exploitation, Karlsruhe. His research interests include Web-based decision support systems for risk and crisis management semantic Web and sensor Web enablement technologies.



RAN TAO received the B.Sc. degree in telecoms engineering from the Beijing University of Post Telecoms, and the M.Sc. degree in telecoms from the Queen Mary University of London (QMUL). He was a Research Assistant with QMUL from 2010 to 2014. His research interests include resilient, scalable message exchange, distributed computing, cloud computing, and Internet of Things.



OCAL NECMIOĞLU received the Ph.D. degree from Boğaziçi University, Turkey. His research interests are tsunami early warning, tsunami hazard, and forensic seismology. He is a member of the Intergovernmental Coordination Group for the Tsunami Early Warning and Mitigation System in the North-Eastern Atlantic, the Mediterranean and Connected Seas (ICG/NEAMTWS) Steering Committee, and the UNESCO IOC Working Group on tsunamis and other hazards.



ULRICH BÜGEL received the degree in computer science from the University of Karlsruhe, Germany. He is currently a Senior Scientist with the Fraunhofer Institute of Optronics, System Technologies and Image Exploitation, Karlsruhe. His research interests include semantic modeling and leveraging semantic technologies for risk management and environmental information systems.