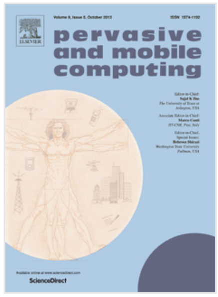
Accepted Manuscript



Semantic web technologies in pervasive computing: A survey and research roadmap

Juan Ye, Stamatia Dasiopoulou, Graeme Stevenson, Georgios Meditskos, Efstratios Kontopoulos, Ioannis Kompatsiaris, Simon Dobson

PII: S1574-1192(14)00198-9
DOI: http://dx.doi.org/10.1016/j.pmcj.2014.12.009
Reference: PMCJ 567

To appear in: *Pervasive and Mobile Computing*

Received date: 8 April 2014
Revised date: 19 December 2014
Accepted date: 25 December 2014

Please cite this article as: J. Ye, S. Dasiopoulou, G. Stevenson, G. Meditskos, E. Kontopoulos, I. Kompatsiaris, S. Dobson, Semantic web technologies in pervasive computing: A survey and research roadmap, *Pervasive and Mobile Computing* (2015), http://dx.doi.org/10.1016/j.pmcj.2014.12.009

Semantic Web Technologies in Pervasive Computing: A Survey and Research Roadmap

Juan Ye[a], Stamatia Dasiopoulou[b], Graeme Stevenson[a], Georgios Meditskos[b], Efstratios Kontopoulos[b], Ioannis Kompatsiaris[b], Simon Dobson[a]

[a] *University of St Andrews, UK*
juan.ye@st-andrews.ac.uk
[b] *Information Technologies Institute, CERTH, Greece*
skontopo@iti.gr

Abstract

Pervasive and sensor-driven systems are by nature open and extensible, both in terms of input and tasks they are required to perform. Data streams coming from sensors are inherently noisy, imprecise and inaccurate, with differing sampling rates and complex correlations with each other. These characteristics pose a significant challenge for traditional approaches to storing, representing, exchanging, manipulating and programming with sensor data. Semantic Web technologies provide a uniform framework for capturing these properties. Offering powerful representation facilities and reasoning techniques, these technologies are rapidly gaining attention towards facing a range of issues such as data and knowledge modelling, querying, reasoning, service discovery, privacy and provenance. This article reviews the application of the Semantic Web to pervasive and sensor-driven systems with a focus on information modelling and reasoning along with streaming data and uncertainty handling. The strengths and weaknesses of current and projected approaches are analysed and a roadmap is derived for using the Semantic Web as a platform, on which open, standard-based, pervasive, adaptive and sensor-driven systems can be deployed.

Keywords: Ontologies, Pervasive Computing, Streaming Query, Uncertainty reasoning, Context Awareness

1. Introduction

Nowadays computing is pervasive, anywhere and any-time, leading to a profound impact on everyday life. To begin with, our homes have installed energy monitoring sensors and intelligent systems that automatically adjust and control heaters based on our behaviours or preferences, aiming at maximising comfort but also minimising energy consumption [75]. Furthermore, smart home based pervasive assistants can help elderly people lead independent lives by collecting symptoms-related data and adapting to different types of physical and cognitive deficits [20, 32]. Even our phones have evolved into a hub of sensing, computing and communication, helping us locate featured restaurants, plan a day trip or suggest a transport mode depending on traffic conditions and weather information.

Moving beyond individuals, by collectively analysing GPS data acquired from a large number of users we can identify hot zones within a city, further contributing to tourist recommendations [159] and urban planning [48]. Outside the cities, sensors can be deployed in remote areas to monitor pollution in landfilled sites [121] or gather environmental information, in order to accurately control the concentration of fertiliser in soil [54]. Such remote sensing supports the long-term collection of fine-grained environmental data, which would be otherwise difficult or impossible to gather. This is a vital factor in fostering scientific research and increasing understanding of the environment and all these broad application scenarios are well supported by the advances in pervasive sensing and communication technologies.

On the other hand, pervasive computing faces the challenge of how to model and reason on such massive amounts of data and how to facilitate sharing and interoperability across heterogeneous systems and applications. For example, how can an intelligent traffic control system effectively use pollution data monitored in a city, in order to design a pollution-free route? Or how can a smart home energy system meaningfully use traffic information, in order to predict when a user will arrive home? Consequently, there is a pressing need for open and standards-based representations that will facilitate integrating information of heterogeneous types and modalities, as well as communicating and exchanging information between devices and components [153].

A suitable solution lies in *Semantic Web* (*SW*) technologies [11], which aim to bring to the table the ability to formally capture intended semantics and to support automated reasoning, supporting sharing, integration and management of information from heterogeneous sources. These capabilities satisfy perfectly all those common requirements in pervasive, sensor-driven and adaptive computing environments mentioned above. Via explicitly rendering meaning, the Semantic Web tries to facilitate data exchange between systems and components in an open, extensible manner, maintaining semantic clarity across applications. SW technologies have demonstrated to successfully address several pervasive computing concerns in a number of small-scaled and targeted applications, such as representing complex sensor data [93], recognising human activities [27] and modelling and querying location data across heterogeneous coordinate systems [131]. Additionally, the use of ontologies elegantly supports the cooperation of data sources within an open system [129], while ontological reasoning proves useful in manipulating structured conceptual spaces [12, 153].

However, the potential of SW technologies in addressing other key pervasive computing application requirements is yet to be fully explored. Sensor data typically exhibit heterogeneous modalities and formats, real-time updating and imperfection. Such data often need to be continuously queried, aggregated to a more consistent conclusion and abstracted to different levels of generalisation for different applications types. In addition, processing and reasoning on such data are often conducted on resource-constrained devices. Thus, the key challenges include representing and reasoning on information uniformly across various sensing technologies, applications, systems and platforms; capturing temporal semantics of data and querying and applying different reasoning schemes to highly dynamic data; and, reasoning in the presence of extensive uncertainty. This survey explores these issues and seeks to answer three questions: (a) to what extent do existing SW technologies address the requirements? (b) what additional techniques might be needed? (c) how might the research community address these deficiencies?

The discussion is organised as follows. The background of SW technologies is described in section 2. Section 3 introduces modelling information and their semantic relations at different levels of abstraction, including raw sensor data, well-structured domain information (*context*), and an atomic concept indicating a change of state (*event*). Section 4 introduces different reasoning mechanisms applied. Section 5 discusses existing strategies in modelling temporal information, facilitating querying on dynamic data and reasoning on temporal knowledge, while section 6 discusses the approaches of handling uncertainty. Finally, section 7 identifies challenging research issues that still require further exploration and section 8 concludes the paper.

2. Background of Semantic Web Technologies

The Semantic Web [11] is a resource-oriented extension of the current Web that aims at a common framework for sharing and reusing data across heterogeneous applications and systems. The rationale is to convert the currently unstructured and semi-structured collection of Web documents into a '*web of data*', where the underlying semantics are expressed in a formal and machine-understandable way. Within this vision, *ontologies* play a key role, providing consensual and formally-defined terms for describing resources in an unambiguous manner.

In 2004, the *Web Ontology Language* (*OWL*[1]) became a W3C recommendation, paving the way for a new generation of state of the art tools (ontology editors and reasoners) and the proliferation of ontology-based applications in several domains. Formally founded on *Description Logics* (*DL*) [6], OWL is endowed with expressive representational constructs that allow capturing complex knowledge. At the same time, OWL avails of well-defined DL reasoning services for affording automated reasoning support. These advantages furnish OWL with a variety of appealing features within the context of pervasive applications. For example,

[1] `http://www.w3.org/TR/owl-features/`

in OWL one can effectively model and reason over taxonomic knowledge. This is a desirable feature in pervasive applications, where there is the need for modelling information at different levels of granularity and abstraction that will drive the derivation of further successively detailed contexts. Similarly, OWL supports consistency checking, another useful feature when dealing with imperfect context information coming from multiple sources.

2.1. Resource Description Framework (RDF)

The *Resource Description Framework* (*RDF*[2]) provides a directed graph formalisation, with nodes representing resources and arcs representing properties. Its semantics are prescribed by two ontology languages: *RDF Schema* (*RDFS*) and OWL. RDFS provides a basic vocabulary for dividing RDF resources into classes and introduces *subClass* and *subProperty* for capturing relations between classes and properties at varying levels of abstraction. On the other hand, OWL, as discussed later, provides a richer ontology language that supports expressing functional, transitive and inverse properties, equivalent properties and classes, and cardinality restrictions on the structure of class members.

For sensor-driven systems, the benefits of these technologies emerge directly from their *formality*. RDF's use of Uniform Resource Identifiers (URIs) in identifying concepts and properties, combined with OWL's support for modelling equivalent classes and properties, allows determining whether lexically identical terms share the same meaning, or if two lexically different terms are synonyms or not. This formality has several advantages: (a) there is no single authority responsible for engineering ontologies or producing data; (b) entities may be described by combining concepts from different ontologies; (c) combining both ontologies and data from multiple sources is straightforward.

Another benefit is *domain-neutrality*. RDF supports the representation of information across disparate application domains, unifying all data under a single model. Also, data across different components in a system and across different systems is seamlessly merged [26]. This can be contrasted to traditional database schemata, where terms and relations have no prescribed semantics, and XML Schema, which is concerned with the hierarchical structure of data elements and not with capturing the underlying relations.

Technologies for managing RDF stores exist in the form of *SPARQL*[3] and *SPARQL Update*. SPARQL supports queries consisting of triple patterns, conjunctions, negations and disjunctions, while SPARQL Update supports the conditional insertion and removal of triples from an RDF store. Similarly to relational databases, there exist various tools supporting RDF-graph level manipulations or providing services to map RDF concepts to programming language type systems. Some representative examples are: *Jena* [84], *OWL API* [66] and *RDFReactor* [147].

A further RDF benefit is that an environment model can build upon, and interlink with, existing ontology-based domain knowledge through the principles of *Linked Data* - a set of best practices for exposing, sharing, and connecting pieces of knowledge on the SW [16]. The premise of Linked Data is that by using URIs and RDF to link to data sources, a semi-structured web of ontologically-represented data emerges that can be navigated and explored. Thus, Linked Open Data provides a structured means for accessing data from existing 'non-pervasive' sources and integrating it with existing RDF representations (or via an RDF wrapper for legacy data sources). This has particular potential for bootstrapping systems with the knowledge held by myriad social information sources on the web [117, 127].

2.2. OWL and OWL 2

OWL's design is strongly influenced by Description Logics (DL) [6]. DLs are a family of knowledge representation formalisms characterised by logically grounded semantics and well-defined reasoning. The main building blocks are *concepts* representing sets of objects (e.g. `Person`), *roles* representing relationships between objects (e.g. `worksIn`), and *individuals* representing specific objects (e.g. `Alice`). Starting from *atomic* concepts, such as `Person`, arbitrary complex concepts can be described through a rich set of *constructors* that define the conditions on concept membership. For example, the concept $\exists$`hasFriend.Person` describes all those individuals that are friends with at least one person.

[2] http://www.w3.org/RDF/
[3] http://www.w3.org/TR/sparql11-query/

OWL comes in three dialects of increasing expressive power: OWL Lite, OWL DL and OWL Full. The first two languages can be considered as syntactic variants of the $\mathcal{SHIF}(\mathcal{D})$ and $\mathcal{SHOIN}(\mathcal{D})$ DLs, respectively. The third and most expressive language is designed to provide full compatibility with RDFS. It neither imposes any constraints on the use of OWL constructs, nor lifts the distinction between instances (individuals), properties (roles) and classes (concepts). However, the price for this high degree of expressiveness is the loss of decidability that makes the language difficult to implement. As a result, the focus lies on the two decidable dialects, and particularly on OWL DL.

Nevertheless, despite the rich primitives provided for expressing concepts, OWL DL has often proven insufficient to address the needs of practical applications [55]. Furthermore, OWL can model only domains where objects are connected in a tree-like manner [89]. This constraint can be restrictive for real-world applications, including the pervasive domain that requires modelling more generic relational structures [154]. Responding to these as well as to other drawbacks concerning the use of OWL in different application contexts, the W3C working group has introduced *OWL* 2 [55]. OWL 2 (equivalent to the $\mathcal{SROIQ}(\mathcal{D})$ DL) is a revised extension of OWL, now commonly referred to as OWL 1 . OWL 2 extends OWL 1 with qualified cardinality restrictions; hence for example, one can assert that a social activity is an activity that has more than one actors: $\texttt{SocialActivity} \sqsubseteq \texttt{Activity} \sqcap \geq 2\ \texttt{hasParticipant.Person}$. Furthermore, it is possible to define properties to be reflexive, irreflexive, transitive, and asymmetric, and to define disjoint pairs of properties, thus providing extended support for capturing mereology relations (i.e. the study of part-whole relations). Three profiles, namely OWL 2 EL, OWL 2 QL and OWL 2 RL, trade portions of expressive power for reasoning efficiency, targeting different application scenarios.

Another prominent OWL 2 feature is the extended relational expressiveness provided through the introduction of complex property inclusion axioms (property chains). To maintain decidability, a regularity restriction is imposed on such axioms disallowing the definition of properties in a cyclic way. Hence, one can assert the inclusion axiom $\texttt{locatedIn} \circ \texttt{containedIn} \sqsubseteq \texttt{locatedIn}$, making it possible to infer that if a person is located, for example, in the Engineering Department of the University, then she is located in the University as well.

2.3. Reasoning

Besides formal semantics, DLs come with a set of powerful reasoning services that are based on efficient, sound and complete reasoning algorithms, with well-understood computational properties (e.g. tableaux-based algorithms [7]). State of the art implementations include *Racer* [60], *Hermit* [92], *Pellet* [124] and *Fact++* [144]. DL reasoning services typically include *subsumption*, *satisfiability*, *consistency*, *instance checking* and *realisation*. Through *subsumption* one can determine whether concept A subsumes concept B, i.e. whether description of A is more general than the description of B, deriving the implicit taxonomic relations among concepts, for example that `Room` subsumes `OccupiedRoom`. *Satisfiability checking* leads to identifying concepts for which it is impossible to have members under any interpretation; a sample unsatisfiable concept, though trivial, is $\texttt{OccupiedRoom} \sqcap \neg\texttt{OccupiedRoom}$. *Consistency checking* allows identifying whether the set of assertions comprising the knowledge base is admissible with respect to the terminological axioms. For example, if `EmptyRoom` and `OccupiedRoom` are asserted as disjoint concepts, then the presence of both `OccupiedRoom(kitchen)` and `EmptyRoom(kitchen)` leads to inconsistency. *Instance checking* denotes the task of determining whether a specific individual is an instance of a given concept, whereas *realisation* returns all concepts from the knowledge base that a given individual is an instance of.

Falling under the *Classical Logics* paradigm, reasoning in DL (and hence in OWL) adopts the *open-world assumption*. Intuitively, open-world semantics assumes that we do not have complete information about the world, providing an elegant way of modelling incomplete information. This assumption is well-suited for sensor-driven systems, where information may be incomplete due to sensor inaccuracies or imperfect observations. For example, if the only available knowledge regarding the residents of a house is the assertion `livesIn(Alice,house)`, we cannot deduce based on it alone that no one else lives in the house. In contrast, formalisms adhering to the *closed-world assumption* make the common-sense conjecture that all relevant information is explicitly known, so all unprovable facts should be assumed not to hold. In our example, this would lead to the conclusion that `Alice` is the sole resident of this house.

2.4. Combining Ontologies and Rules

To achieve decidability, OWL trades expressiveness for reasoning efficiency. The tree-model property, mentioned above, is one such example; as a result, it is not possible to describe classes whose instances are related to an anonymous individual through different property paths . To leverage OWL's limited relational expressiveness and overcome modelling shortcomings that OWL alone would insufficiently address, research has been devoted to the integration of OWL with rules. User-defined rules on top of the ontology allow expressing richer semantic relations that lie beyond OWL's expressive capabilities and couple ontological and rule knowledge.

A proposal towards this direction is the *Semantic Web Rule Language* (*SWRL*) [67], in which rules are interpreted under the classical first order logic semantics. Via allowing concept and role predicates to occur in the head and body of a rule, SWRL maximises the interaction between OWL and rule components, but at the same time renders the combination undecidable. To regain decidability, proposals have explored syntactic restrictions on rules [91, 116] as well as their expressive intersection [56]. The DL-safe rules introduced in [91] impose the application of rule semantics only over known individuals. It is worth noting that in practice DL reasoners providing support for SWRL actually implement a subset of SWRL based on this notion of DL-safety. Parallel to these efforts, a highly challenging and active SW research area addresses the seamless integration of open and closed world semantics. Representative initiatives in this quest include the hybrid formalism of *Minimal Knowledge and Negation as Failure* (*MKNF*) knowledge bases [90], the extension of ontologies through the use of *integrity constraints* and the *grounded circumscription* approach.

Taking a different perspective, a number of approaches have investigated combining ontologies and rules based on mappings of a subset of the ontology semantics on rule engines. For instance, [139] defines a weakened variant of OWL Full, according to which classes can also be instances and are extended to apply to a larger subset of the OWL vocabulary. Inspired by the previous approach and DLP [56], the semantics of the OWL 2 RL profile is realised as a partial axiomatisation of OWL 2 semantics in the form of first-order implications, known as OWL 2 RL/RDF rules. Especially for the case of sensor-driven systems in the pervasive domain, expressing rich semantic relations is essential. The reason lies in the fact that the derivation of high-level knowledge from low-level sensor data requires relational structures that capture the interrelation of various pieces of information in terms of time, location, actors and resources.

2.5. Summary

This section has introduced the basic notions underlying the Semantic Web and provided a brief overview of key technologies empowering the envisaged knowledge sharing and reuse across heterogeneous environments. Expressive ontology languages allow the elegant capture of complex knowledge and its semantics in a formal way, rendering it amenable to automated reasoning tasks with well-understood computational properties. Rules augment further the expressive capabilities, by allowing the representation of richer semantic relationships. Table 1 summarises the advantages and disadvantages of the formalisms presented in this section with respect to reasoning complexity, expressivity and suitability in application domains.

Given the inherently open nature of pervasive, sensor-driven systems, where a crucial requirement is the need to aggregate low-level context information and meaningfully integrate domain knowledge, it comes as no surprise that SW technologies have been acknowledged as affording a number of highly desirable features.

3. Modelling Sensor Data, Context and Events

Information within a pervasive sensor-driven system may include raw sensor data, context, domain knowledge and events. Raw sensor data can be abstracted into relations of well-structured concepts, called *context*, which describe properties of an environment or a user, such as *Location*, *Time*, *Person* and *Resource*. Context associates concepts in each domain with particular properties and relationships and, thus, provides a uniform way of representing sensor data. This makes the latter sharable and re-usable between different systems, regardless of the heterogeneity and complexity of the underlying sensing technologies. If we consider a context representing a state of an environment, an *event* indicates a change in the state that should be identified, processed and managed by the system, in order to deliver personalised services to users; e.g. an

Table 1: Advantages and disadvantages of the formalisms presented in Section 2

	Advantages	Disadvantages
RDF	reasoning efficiency	limited expressivity
OWL Full	highest expressivity	non-decidable
OWL DL	high expressivity	high computational complexity
OWL 2 DL	higher expressivity than OWL DL	high computational complexity
OWL 2 EL	low computational complexity, handles efficiently ontologies with large number of classes and properties	low expressivity
OWL 2 QL	handles efficiently very large volumes of instance data, fast query answering	moderate expressivity
OWL 2 RL	scalable reasoning without sacrificing too much expressive power, can be easily implemented using a standard rule language	less scalable than EL
SPARQL	W3C standard, supported by the majority of RDF repositories	limited availability of tools for supporting users (e.g. SPARQL query editors)
SWRL	high expressivity (higher than OWL DL)	not a standard, non-decidable
DL-Safe rules	decidable, supported by reasoners	less expressive than SWRL (handles only known individuals)

event of 'a user having a heart attack' might trigger an application of calling an ambulance. An event can be conceptually defined as an action or occurrence that happens at a certain time within an environment and is described through basic elements such as *when*, *where*, *what*, and *who*.

Providing a commonly-agreed vocabulary to represent sensor data, context and events is important for making them understandable, sharable and interoperable across different systems and platforms. Such a vocabulary needs to be sufficiently rich to represent properties, structures and relationships among information, thus, facilitating querying, personalising and further processing of data, as well as their sharing and reuse. Additionally, this common vocabulary can be used for semantically annotating data and 'meaningfully' relating it to other data. In the following sections we will discuss to what extent SW technologies have contributed to providing such a vocabulary to model sensor data, context, and events.

3.1. Sensor and Sensor Data modelling

Pervasive sensors have exhibited heterogeneity in multiple aspects in that they produce different values, with different data schemas, precision or accuracy, and in different units of measurement [23]. Heterogeneity leads to significant difficulty in integrating and querying data over multiple sensor networks. The main contributions of SW technologies to modelling sensor data are providing uniform syntactic representations, enhancing their semantic meaning from their associated domain concepts, reasoning over sensor data to derive new knowledge, and facilitating querying over live sensor streams. The following subsections focus on the first two topics, while reasoning is further discussed in section 4.

3.1.1. Uniform Syntactic Representation

There exist small-scale sensor ontologies serving as uniform syntactic representations for a particular domain, for example *Sensory Data Set Description Language* (*SDDL*) [68] and ontologies in $\mathcal{PI}$ [156] for representing data sets collected from sensors deployed in smart home environments. These ontologies mainly aim towards facilitating sharing and reusing data between researchers. SDDL is an XML-encoded description language for specifying sensors, actuators, data set parameters, and sensor events. $\mathcal{PI}$, modelled in OWL, represents sensor observations and links each field to the domain ontologies; for example *Location*, *Domain* and *Object*.

Each sensor type is associated with a *domain* that describes what is measured (e.g. acceleration or gas usage), and a technical specification that includes its *manufacturer*, *model*, *size*, *deviation* of readings (e.g. 'a maximum precision of ± 1.5% full scale' for a gas flow sensor), its sampling *frequency* and a number of *valueRange* parameters – boundary values that characterise different states. These values may come from the manufacturer or through the application of learning techniques. It supports deriving high-level context from raw sensor readings; e.g. a context `stoveOn` is inferred if the gas sensor on the stove has a reading greater than a threshold.

```
[stoveOnRule:
       (?A sensor:domain "GAS"),
       (?A sensor:value ?B),
       (?A sensor:object ?C),
       (?C rdf:type :Stove),
       ge(?B "1"^^xsd:double)
-> createClassification(?A, "stoveOn")]
```

A more generic framework for sensor data modelling and encoding is the *Sensor Web Enablement* group (*SWE*) [136] proposed by the *Open Geospatial Consortium* (*OGC*). SWE specifications include three core languages: *Sensor Model Language* (*SensorML*), *Observations & Measurements* (*O&M*) and *Transducer Model Language* (*TransducerML*). These languages provide standard models and XML schemas for encoding sensors, sensor networks, processes, and sensor observations, etc. These specifications offer the following advantages: (a) a standardised communication and interaction with arbitrary types of sensors and sensor systems; (b) availability of sensors over the Web through well-defined formats and Web service interfaces; (c) concealment of the sensor communication details and the heterogeneous sensor protocols from applications working on top of these services [22]; (d) standardisation of the discovery, exchange and processing of sensor observations [21]. However, these specifications are mostly based on XML data, which lacks the support of semantic interoperability and of linking the described resources to the existing knowledge [150].

3.1.2. Semantics Enhancement

In order to extract new knowledge from raw sensor data, we need to enhance the latter with semantics. This research has been developed through several stages, from relating sensor ontologies to domain ontologies [119], to focusing on analysing spatial, temporal and thematic semantics of sensor data [150] and to applying the Linked Data principle [70] (see section 2.1) when more and more public ontological sources are available. *OntoSensor* [119] is one of the early works, built on SensorML (see 3.1.1), the IEEE *Suggested Upper Merged Ontology* (*SUMO*)[4] and ISO 19115. It is a comprehensive deep sensor ontology, which includes a domain theory expressed in a language constructed using the functional and relational basis sets to support ontology-driven inference. It is able to capture sensors' computing and communication capabilities and their percept attributes that are used to store measurements of physical phenomena, and detection, classification and tracking of physical objects. OntoSensor also includes more advanced inference mechanisms that can be used for synergistic fusion of heterogeneous data.

Semantic Sensor Web (*SSW*) [123] enhances the meaning of sensor observations by adding semantic annotations to SWE standards (see previous subsection), such as *spatial*, *temporal* and *thematic* semantics. The spatial metadata refers to sensor location and data in terms of a geographical reference system, local reference, or named location. The temporal metadata refers to the time instant or interval when the sensor data is being captured, while the thematic metadata describes a real-world state extracted or abstracted from sensor data.

Following SSW's idea of semantic annotation, the Linked Data principle presents a more generic approach by creating RDF links between sensor data and concepts on the Semantic and Social Web [70]; that is, the concepts published by authoritative sources (e.g. DBpedia) or user-generated content (e.g. tags) on the Social Web. Such annotation enables reasoning over the sensor and the linked data to provide advanced

[4] `http://suo.ieee.org/SUO/SUMO/SUMO_173.kif`

sensor data query and retrieval functions. Janowicz et al. [70] present a Linked Data model and a RESTful proxy for the OGC Sensor Observation Service to improve integration and interlinking between sensor observations for the digital Earth. This RESTful proxy is used to assign meaningful identifiers to sensor data and to directly publish the raw data on the web. A Semantic Enablement Layer is implemented to encapsulate SW reasoners and repositories within the OGC services (see 3.1.1) and thus, to enable a transparent and seamless integration of SW technologies with the Spatial Data Infrastructure.

More sensor ontologies can be found in relevant reviews by W3C and Compton et al [31]. The W3C Semantic Sensor Network Incubator group developed the *Semantic Sensor Network* (*SSN*) ontology [30]. It provides a formal and machine-processable representation of sensor capabilities, properties, observations and measurement processes, to aid in searching, querying, and managing sensor networks and their data. Central to the ontology is the *Stimulus-Sensor-Observation* (*SSO*) ontology design pattern that provides a lightweight model for representing sensors, their inputs (called stimuli) and observations. SSO is re-usable for a variety of application areas and it can be used in conjunction with other relevant ontologies. Both SSN and SSO have been aligned with the DOLCE+DnS Ultralite (DUL) ontology [51], in order to facilitate the integration into more complex ontologies as a common ground for alignment, matching, translation, and interoperability.

3.1.3. Analysis

Table 2: Sensor Data modelling

Requirement		Techniques
Syntax	Light-weight, small-scale	[68, 156]
	all-encompassing, standard	SWE [136]
Semantics	Relating to domain ontologies	[119]
	Semantic annotation	SSW [123]
	Linked Data	[70, 150]

Conclusively, the sensor- and observation-centric ontologies presented above have evolved from a syntactic to a semantic and knowledge model. Table 2 presents an overview. SDDL and $\mathcal{PI}$ tend to share and reuse sensor data sets by describing sensors and data in a light-weight syntactic model. $\mathcal{PI}$ supports classifying raw sensor data into high-level context through user-specified rules. SDDL and $\mathcal{PI}$ focus on data and are relatively easy to adopt, while more standard solutions like SensorML aim to standardise interfaces for services and description languages for sensors and their processes to enable syntactic interoperation.

At the same time, various sensor ontologies have been developed to derive knowledge from raw sensor data. One of the first initiatives, OntoSensor, supports ontology-driven inference by combining the domain conceptual model with the syntactical standards. SSW applies a more generic analysis on semantic dimensions of sensor data, distinguishing spatial, temporal, and thematic aspects. The principle of Linked Data seems as a promising approach to enhance semantics of raw sensor data by linking them to domain concepts on other standard resources. In fact, the generic ontologies like SensorML and SSN must be reconceptualised, re-defined, combined, or extended with other domain ontologies, so that they can be reused in a particular application domain. In summary, sensor ontologies are moving towards a deeper knowledge model to express and automatically create a composition of processes of sensors and sensor data.

3.2. Context modelling

The applications encompassed by the broad notion of pervasive computing are vastly heterogeneous in nature and broad in their data requirements and scope. Hence, in an open environment it is impossible to define a 'complete' model of content, without *a priori* knowledge of the applications that will use it. However, a large number of applications exhibit overlapping data requirements, the most common of which are the need to represent time, location, actors and resources. These neatly correspond to the notions of *when*, *where*, *who*, and *what*.

3.2.1. When - Time

Temporal features play a key role in enhancing entity descriptions within a pervasive environment. For example, representing the time at which an event takes place, the *frequency* with which a sensor samples the environment (e.g. every 10 seconds), or the expected *duration* of an activity (e.g. 30 minutes). Here, we must also consider semantically meaningful notions of time, which may have an exact (e.g. Monday morning, Christmas) or fuzzy (e.g. lunch time, trip duration) correspondence to physical time.

The most common representation for physical time is the ISO 8601 standard [69], which is based on the Gregorian calendar, and provides a lexical format for modelling dates, times, durations, time intervals, and time zones. For example, `2012-05-10T09:32BST` represents the time of 9:32am on the 10th of May, 2012 in British Summer Time. A subset of the lexical formats defined in this standard is adopted by XML schema, and is further adopted by RDF as a means for typing date literals. Given any two temporal features, there exists a relationship between them that we may also wish to model. Instants are related to intervals by the notion of *containment*, while Allen's temporal calculus [1] defines the following seven relationships between time intervals

- *during(t1, t2)*: time interval *t1* is fully contained within *t2*;
- *starts(t1, t2)*: time interval *t1* shares the same beginning as *t2*, but ends before *t2* ends;
- *finishes(t1, t2)*: time interval *t1* shares the same end as *t2*, but begins after *t2* begins;
- *before(t1, t2)*: time interval *t1* is before interval *t2*, and they do not overlap in any way;
- *overlap(t1, t2)*: interval *t1* starts before *t2*, and they overlap;
- *meets(t1, t2)*: interval *t1* is before interval *t2*, but there is no interval between them, i.e. *t1* ends where *t2* starts;
- *equal(t1, t2)*: *t1* and *t2* are the same interval.

and their inverses (*contains*, *startedBy*, *finishedBy*, *after*, *overlappedBy*, and *metBy*) for a total of thirteen relationships (equals being its own inverse).

The W3C *OWL-Time ontology*[5], which provides a vocabulary for Allen's thirteen temporal relations, uses XML Schema's `dateTime` formats, but also provides its own component based on `DateTimeDescription` that can express additional information (e.g. 'day of week' and 'day of year') for facilitating the mechanical extraction of and reasoning on this information. However, note that as a temporal relationship is formed by each pair of temporal entities, it is impractical to manually specify or concretely realise these in any data model of notable size. This indicates a need for special consideration when evaluating these temporal predicates as part of a query.

Standard Ontology for Ubiquitous and Pervasive Applications (*SOUPA*) [25], one of the well-known ontologies in pervasive computing, supports a formal way to model context and thus provides rich semantics for programming. It borrows terms from other standard domain ontologies such as FOAF[6], DAML-Time [65], OpenCyc [80], RCC [19], and the Rei Policy [73] Ontology. SOUPA's temporal predicates are based on DAML-Time (predecessor of OWL-Time). On the other hand, *Ontonym* [129], a set of upper ontologies that represent core concepts in pervasive computing, adopts OWL-Time directly in its modelling of temporal concepts. Temporal concepts may be directly mapped to the top level ontologies discussed. Although it is possible to represent all the temporal relations, only a subset of the relations map to the properties of the underlying model (e.g. during or overlap) upon which standard inference is performed.

[5]`http://www.w3.org/TR/owl-time/`
[6]`http://xmlns.com/foaf/spec/`

3.2.2. Where - Location

Location information provides a means of invoking application behaviour based on the real world positioning of users and artefacts. Location information has a number of possible representations, ranging from *absolute*, to *relative*, and to *symbolic*, all of which can be related, and with specific forms more appropriate than others for any given application [42]. Location is often regarded as the most important type of context, and related models and query frameworks have been already extensively researched, before the emergence of the Semantic Web (e.g. [71, 108]).

The *CONtext ONtology* (*CONON*) [57] is a context model that encompasses a common upper ontology for the general concepts in pervasive computing as well as domain-specific ontologies that apply to different subdomains like smart homes. *Aspect-Scale-Context* (*ASC*) is a contextually based model [134]. A context is a set of contextual information characterising entities (like a person, place, or a general object) relevant for a specific task in their relevant aspects. An aspect is a classification, symbol- or value-range, whose subsets are a super-set of all reachable states, grouped in one or more related dimensions called *scales*. For example, 'GeographicCoordinateAspet' is a location aspect, which may have two scales 'WGS84Scale' and 'GaussKruegerScale'. A piece of contextual information is an object instance under a certain scale, e.g. `new GaussKruegerCoordinate("367032", "533074")`. As presented, the ASC's location ontologies are mainly used for modelling spatial positions in these coordinate systems and for representing distances between positions. In contrast, the primary goal of the CONON location ontologies is to differentiate between indoor and outdoor spaces.

More comprehensive is SOUPA's location ontology, which is formed through the conjunction of two existing location vocabularies, *OpenCyc* [80] and *Region Connection Calculus* (*RCC*) [19]. OpenCyc supports the symbolic representation of spaces, while RCC provides a set of spatial relations (e.g. overlap, disconnection, and tangental part) that may hold between two regions – essentially the location equivalent of Allen's temporal relations discussed above. *Ontonym* adopts a simpler spatial relation model, supporting only containment, overlap, and adjacency. It supports multiple coordinate systems (based on the coordinate translation scheme of Jiang et al. [71]) and incorporates a notion of relative positioning between entities based on compass directions and distance. An API (Application Programming Interface) is also provided for querying the position of entities and for generating paths between two points in the model [131].

3.2.3. Who - Person/Agent

This context type broadly relates to the actors in a pervasive system, which are typically people or agents whose identities are manifest by software. The data representation requirements associated with these entities are usually application-specific, and focus on the role played, or interactions expected of an entity in a scenario. The CONON person ontology is tightly-coupled to its application offerings and provides a small vocabulary for describing a person's name, age, and situation. FOAF is an ontology centred on linking people not only through social relationships, but also networks of human collaboration and interests. SOUPA builds upon FOAF to support the expression of human profile information (name, age, and contact details), and *MoGATU BDI* [104] to support the description of agent state – their beliefs, desires, and intensions. On the other hand, Ontonym's person vocabulary is based on a subset of terms from *vCard*[7], W3C *PIM*[8] and FOAF, and provides support for modelling date of birth, gender, language, and contact profiles, postal and email addresses, telephone and fax numbers, and web presence. Ontonym provides a component-based name model that supports the semantic modelling of name terms (e.g. `ProfessionalTitle`, `GivenName`, and `PatronymicName`), and borrows from Davies and Vitiello's relationship vocabulary for FOAF to define social relations between people covering genetic, working, romantic, residential, and friendship connections.

A recent trend is towards applications that use the information available about the social relations or shared interests between groups of people – collectively, their *social context* – as a primary driver in adapting application behaviour. Biamino [13] posits that objects in a pervasive environment should have the capability to detect users and the social connections between them, should be able to infer a group's social context

[7] `http://www.w3.org/2006/vcard`

[8] `http://www.w3.org/2000/10/swap/pim/`

according to its network structure (i.e., its size and density of social relationships), and correspondingly provide a context-driven output.

Kourtellis et al. [76], Toninelli et al. [143] and Kabir et al. [72] all develop social context management frameworks, intended as a basis for developing socially-aware applications, that draw data from social media platforms (e.g., Facebook, LinkedIn, and Twitter), with the common aims of providing: *i)* a semantically rich repository of social relation data, *ii)* an ontology-based modelling substrate for representing either "object-centric" relationships (based on common interest), "people-centric" relationships (formal and declarative), or both, *iii)* a uniform API for accessing the social data, and *iv)* socially aware access control to the managed data.

Of these frameworks, the Social Context Information Management System (SCIMS) of Kabir et al. [72] supports both object- and people-centric relations. SCIMS is based on a core upper ontology that defines four first-class entities: `Person`, `SocialRole`, `Relationship`, and `CurrentStatus`, and domain specific ontologies that extend from this. For example, relationship models for `Family`, `Work`, and `CommonInterest`, each with further refinements (e.g, work relations for an educational institution are further specified as `Student-Teacher`, `Student-Supervisor`, or `Colleague`). Role based access control policies use the model concepts to define under which conditions information about a given resource is accessible.

3.2.4. What - Resource

Given that the term resource is itself generic, the lack of an all-encompassing vocabulary for describing resources is not surprising. A resource may refer to anything not otherwise explicitly modelled in a particular pervasive system, for example, a device, an image, a document, a physical object or a virtual artefact. While the properties of a resource might be application-specific in nature, resource descriptions might draw from multiple ontologies. For example, FOAF provides some general terms for describing images, documents, and projects, the *Dublin Core Metadata Initiative* (*DCMI*[9]) vocabulary provides terms for describing aspects of resources such as their creators, representation format, and licensing, and the *GoodRelations* ontology [64] provides an e-commerce vocabulary for describing products and services.

3.2.5. Movement and Trajectory

With the increasing popularity of GPS-equipped devices, it becomes very convenient to collect the positioning data about human, animals, cars and ships. Thus, the research on movement analysis and trajectory discovery have gained lots of attention during the past few years. Various machine learning and statistical techniques have been applied to extract meaningful trajectory patterns, however, often these patterns are either too difficult to explain or understand, or too far away from the real need of applications [152]. Ontologies have been applied to try to compress and abstract raw GPS data into higher-level qualitative information. One strand of research is to represent a trajectory into a sequence of stops and moves [61, 125, 152]. These models focus on stops or points (or areas) of interest; for example, one trajectory can be represented as $\langle Building_1, t_1\rangle \rightarrow \langle Room_1, t_2\rangle \rightarrow \langle Room_2, t_3\rangle$ [152]. With the help of geographic and application knowledge, higher-level application queries can be answered, such as "how many cars visited a gas station today".

Another strand of research is to use association rules to extract dynamic characteristics from raw GPS data, such as acceleration and speed. Based on them, a set of motion patterns can be described such as `stand still` – no perceivable movement, `steady motion` – movement with unchanged speed, `positive acceleration` – movement with increasing speed, `negative acceleration` – movement with decreasing speed, `positive course change` – movement with the course angle changing over a certain degree, and `negative course change` – movement with the angle changing less than a certain degree. With this taxonomy, the raw GPS data can be compressed into a sequence of motion patterns, which will be more straightforward to be used by high-level applications. The computation of these patterns are still based on statistical methods, such as a simple threshold based motion calculation [109] or piecewise linear algorithm [2, 94].

[9] `http://dublincore.org/documents/dcmi-terms/`

3.2.6. Analysis

This section evaluated the way in which the most common elements of context in pervasive computing are represented. While one can adopt to a degree a single, shared temporal and location model, upon which event descriptions are based, the specification of the actors and resources in a pervasive environment still remains highly application-specific. Where the need to model particular features of such entities occurs separately (i.e. using separate vocabularies), the promise of *Linked Data* [16] plays a critical role in supporting the integration of entities with diverse descriptions across applications and environments. Table 3 lists the most commonly adopted ontologies in pervasive communities.

Table 3: Most commonly adopted ontologies in context modelling

Context Type	Ontologies
When	DAML-Time, W3C OWL-Time
Where	OpenCyc, RCC
Who	vCard, W3C PIM, BDI, FOAF
What	FOAF, DCMI

Beyond the above four primitive types of context, we have also introduced a higher-level type of context – movement and trajectory. From the mentioned work, we have seen that there exists few work on providing semantic representation for raw GPS data, which is often too bulky and thus inefficient to process and query. Most semantic work focuses on abstracting the raw movement data into trajectory patterns with qualitative information. Geographic and application knowledge can be applied to enrich the trajectory and thus answer higher-level application queries.

3.3. Event modelling

The efficient representation and processing of events is an important and challenging task in pervasive environments. In most cases, only a small number or a combination of raw primitive events that are generated directly by the sensors or after low-level data processing is of real interest. However, the recognition of high-level, real-world complex events, such as situations and human activities, often requires incorporating expressive domain knowledge that would enable further correlation of events and multi-modal information beyond predefined patterns and attributes.

In order to overcome the above limitations, research efforts have focused on the definition of ontology-based event models. Ontologies are used in this context as common vocabularies for representing knowledge relevant to events and their inter-relationships. At the same time, they assist in solving interoperability problems and providing the means for high-level event interpretations based on ontology reasoners (see section 4). This section reviews existing domain-independent ontologies for event modelling, presenting the different design patterns and expressive capabilities they provide.

Almost all the event ontologies support representing the three key aspects of an event: *when* and *where* an event happens and optionally *who* is participating in the event. Generally an event is defined as a generic concept and links with temporal, spatial, and actor ontologies (as introduced in section 3.2) via corresponding properties. For example, in *Simple Event Model* (*SEM*) [146], the class `sem:Event` is associated with `sem:Actor`, `sem:Place`, and `sem:Time`. Each of these core classes is associated with the `sem:Type` class that is used to aggregate implementations of type systems from other ontologies promoting for re-usability of existing type vocabularies.

An event can also be modelled hierarchically. In SOUPA [25], an event is a generic concept and the spatial and temporal aspects are modelled as a subclass of both `soupa:Event` and `soupa:SpatialTemporalThing`. In Ontonym [129], an event is defined as the union of the `ontonym:InstantEvent` and `ontonym:IntervalEvent` classes that are used to model events that occur instantaneously or over a time period, respectively. These two classes are further extended with location ontologies to represent the spatial aspect of an event; that is, `ontonym:SpatioInstantEvent` and `ontonym:SpatioIntervalEvent`. Their different ways of modelling the *when* and *where* aspects of an event have been depicted in Figure 1.

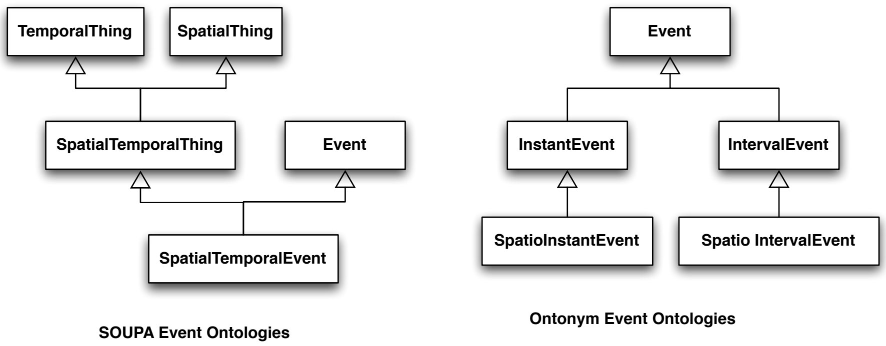


Figure 1: Event Ontologies in SOUPA and Ontonym

Who is modelled when an event involves an active actor which can be a person, device, resource, object, or even an agent. In Ontonym, the property `ontonym:containsRole` identifies the types of roles that can be taken in an event and the `ontonym:playsRole` property is used to associate an entity with an event. The Event Ontology (EO)[10] uses `eo:agent` to represent actively participating agents. There exist rich semantic relationships between events. Event-Model-F follows the descriptions and situations ontology design pattern (DnS) [52], introducing six design patterns for modelling various aspects of events. The *Participation Pattern* models the participation of objects (`dul:Object`) in events (`dul:Event`); the *Mereology Pattern* models the composition of a composite event out of its component events; the *Causality Pattern* expresses relationships between events that play the roles of causes and effects; the *Correlation Pattern* clusters correlated events (i.e. events that have a common cause and there is no causality relationship among them); the *Documentation Pattern* is used to provide evidence for events (evidence may be a specialisation of the `dul:Object` class or another event); and finally, the *Interpretation Pattern* models different viewpoints on which the perception of an event may depend.

The LODE vocabulary is aligned with other event-related vocabularies and ontologies, such as DUL [51], EO and CIDOC [43]. Events are modeled as instances of the `lode:Event` class that is defined as subclass of the `E2_Temporal_Entity` class of CIDOC and is equivalent to the `eo:Event` and `dul:Event` classes. An event can be further associated with a time interval through the `lode:atTime` property that is a subproperty of CIDOC's `P4.has_time-span` and `dul:isObservableAt` properties. The `lode:inSpace` property relates an event to some spatial boundary. The association of events with objects and agents is performed via the property `lode:involved` and its subproperty `lode:involvedAgent`, respectively.

3.3.1. Analysis

Events capture the dynamic aspects of a domain and their efficient representation, processing and analysis are considered key requirements in pervasive and sensor-driven environments. The motivation behind the development and use of ontology-based event models is to provide formal and explicit vocabularies for semantically representing and correlating common aspects of events (e.g. places, people, and objects) amenable to reasoning (see section 4), and thus to high-level interpretation. This section briefly presented existing event ontologies, focusing on the provided ontology constructs and design patterns. The representation of common aspects of events, such as time, location and participation, is supported by all ontologies. However, the representation of more complex event relationships, such as mereological or causal relationships, are fully supported only by the Event-Model-F ontology that provides a rich axiomatisation of ontology design patterns. SOUPA and Ontonym follow a modular design with a moderate axiomatisation compared to EO, LODE and SEM ontologies. Among them, only SEM is capable of capturing different interpretations of the

[10] `http://motools.sourceforge.net/event/event.html`

Table 4: Comparison of the event ontologies

Ontologies	When	Where	Who	Mereology	Causality	Correlation	Interpretation
SOUPA	√	√	√	–	–	–	–
Ontonym	√	√	√	–	–	–	–
Event Ontology	√	√	√	*limited*	*limited*	–	–
SEM	√	√	√	*limited*	–	–	√
Event-Model-F	√	√	√	√	√	√	√
LODE	√	√	√	–	–	–	–

same event, which has potential benefit in supporting multiple applications. Table 4 summarises the strong and weak points of each ontology.

4. Reasoning

The higher-level integration of raw context data as well as the comprehension of their meaning are key prerequisites towards understanding a user's state, behaviour and surroundings. Espousing the ontology-based modelling paradigm, the low-level context information acquired from sensors is translated to respective ontological class and property assertions. Hence, typical ontological reasoning tasks can be employed for checking the consistency of the aggregated set of contextual assertions, and more importantly, to derive more complex context abstractions (e.g. recognise a user's activity based on current location and objects used) that would otherwise remain implicit. In many cases, data-driven approaches are used in the first place, such as machine learning and statistical models, to extract observations from the sensory data collected through the sensor monitoring infrastructure. The data then serve as input to the semantic representation layer, where ontologies provide the common reference point for the projection and aggregation of the extracted descriptions. Finally, the interpretation layer provides the context-aware semantic reasoning procedures for the derivation of higher level context abstractions. Figure 2 presents the abstract architecture of a semantically-enriched pervasive framework.

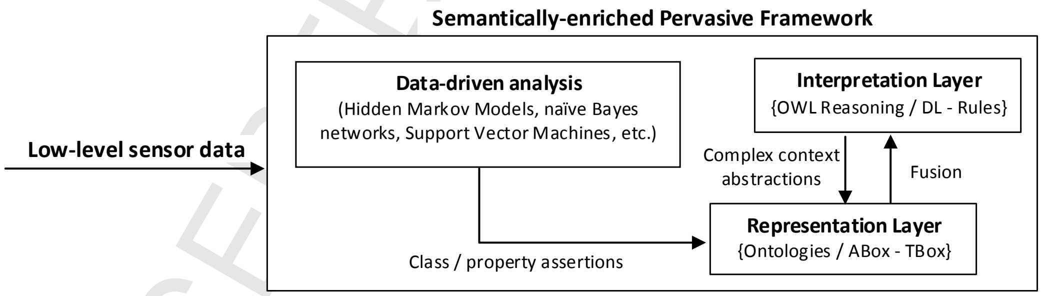


Figure 2: An abstract semantically-enriched pervasive architecture.

In addition to adopting plain ontology-based solutions though, a number of approaches have explored reasoning frameworks that combine ontologies and rules [12] in order to cope with OWL's restricted relational expressiveness (see section 2.4). Based on the level of interaction that different combination frameworks afford, these approaches can be discriminated into three categories: maximising the interaction between the ontology- and rule-based inferences; adhering to a tight integration; and adopting a much looser notion of coupling. In the following, representative frameworks for each of the three paradigms are discussed.

4.1. Ontology-based frameworks

Semantic Smart Home (*SSH*) [26] is one of the characteristic examples using plain ontology-based reasoning. In SSH, activities of daily living (ADLs) are modelled through restrictions in relation to equipment, location and other types of constraints. The inference of ADLs is performed on the basis of the assertional knowledge made available through sensors. For example, an ADL a is inferred to be an instance of `KitchenADL` $\equiv$ $\exists$`hasLocation.Kitchen`, if there is sensor evidence that asserts `hasLocation(`a`, kitchen)`.

Towards a more effective engineering of the domain knowledge, a generic methodology for situation modelling has been proposed in [126]. The methodology is based on the systematic decomposition of situations according to *aspects of interest* that consider spatial, temporal and acting person criteria, and aims to assist system developers in effectively capturing situations at different levels of granularity. Such modelling allows to effectively avail of the subsumption semantics and derive inferences (though at coarser levels of abstraction) when not all pieces of relevant context information are available.

A more recent example is the OWL 2 based context modelling and reasoning architecture presented in [112]. Ontologies are used to represent not only activities but also relevant knowledge that can drive activity recognition, including locations, objects, and so forth. Sensor data are first fed to *COSAR* (context-aware activity recognition system) [111], where primitive activities are derived through a combination of statistical and ontological reasoning. These simple activities, in combination with further contextual data, once aggregated and processed so as to resolve possible conflicts, formulate the assertional knowledge, over which OWL 2 DL reasoning is applied to recognise more complex activities. Via OWL 2's support for composition of properties and for qualified cardinality restrictions, the captured knowledge is considerably more expressive compared to that afforded by earlier frameworks that use OWL DL.

4.2. Tightly-coupled frameworks

An early example of a context reasoning framework that combines ontologies and rules is Gaia [115], an infrastructure for smart spaces – pervasive computing environments that encompass physical spaces. Context information in Gaia is represented as first-order predicates, with the name of a predicate indicating the type of context described. Higher-level contexts can be deduced based on a set of predefined rules that are reevaluated whenever a change occurs. For reasoning in first-order logic, the XSB [120] reasoning engine is used. More recent examples include the context-aware systems [157] that investigate the use of SWRL rules. In particular, OWL and SWRL are used to capture and reason over contextual information about museum visitors, including their preferences and surroundings, in order to provide context-aware recommendation services [157]. Racer and Jess comprise the system's reasoning module, allowing for consistency checking and taxonomic classification (subsumption checking), and the processing of SWRL rules and queries, respectively.

Zhang et al. explore the use of SWRL rules in a SW-enabled framework for self-management in pervasive computing [158]. More specifically, a set of *Self-Management Pervasive Service* (*SeMaPS*) ontologies is used to capture salient notions about persons (including their habits and preferences), locations, software agents, devices, malfunctions and recovery solutions, quality of service parameters and dynamic context aspects, and so on. SWRL rules are used in parallel to capture those parts of complex contextual knowledge that cannot be expressed in OWL. Through the Protege-OWL/SWRL APIs, the RacerPro and Jess reasoning engines are used to derive higher-level context abstractions. Despite the use of SWRL rules, reasoning in both frameworks remains decidable, since the current reasoner implementations employ inherently the DL-safety notions.

4.3. Loosely-coupled frameworks

Unlike the previous frameworks, loosely-coupled frameworks minimise the interaction between ontology- and rule-based reasoning. Rules are applied to the consequences derived by means of ontology reasoning, and affect the knowledge base lightly. Such a framework is the *Service-Oriented Context-Aware Middleware* (*SOCAM*) [58]. In SOCAM, ontology-based reasoning is used to deduce additional knowledge from the context data that are directly acquired through sensors; e.g. based on the transitive semantics of the `locatedIn` relation, the reasoner can infer that a person is located inside the house, provided that she is located in the house's bedroom. Once implicit knowledge is made explicit, first order logic rules are invoked to derive higher-level contexts, such as sleeping, cooking and watching TV. A similar rationale has been adopted in the Semantic Space framework [149]. *RDQL* (*RDF Data Query Language*) queries over the context knowledge base allow to examine desired contexts, so that relevant sets of first order rules can be invoked to derive higher-level contexts.

Ontologies and rules have also been coupled in the health monitoring and alarm management system proposed by Paganelli and Giuli [100]. The context model consists of four ontologies that capture knowledge about patients (e.g. heart rate and body temperature), home environmental parameters (e.g. humidity), alarm management (including policies and contact person information), and social context (e.g. relatives), respectively. Context reasoning is triggered whenever a change in the context knowledge base occurs. Ontology-based reasoning is employed to determine the complex context class to which a specific instance belongs (i.e. realisation) and to check the consistency of the knowledge base once a new conclusion is inserted.

The context-aware access control framework proposed in [142] constitutes another example. The framework follows a hybrid architecture, combining a DL reasoner (Pellet) and a production rule engine (Jess) in order to apply more expressive context reasoning such as deriving property path relationships. For instance in a meeting situation, given that the owner of a requested resource is located at the same place as the resource requestor, we could infer that these two persons are co-located. Such relationships can be expressed in terms of production (if-then) rules whose head and body match classes and properties of the ontology.

However, the close-world semantics that is usually employed for rules may easily lead to incorrect inferences. Such semantics allow the rules to introspect the knowledge base and derive conclusions based on the absence of information. This induces a non-monotonic behaviour, where new inferences may invalidate previously derived conclusions. A representative example is given in [112], which considers a three-room smart home, equipped with four sensors, one in each room monitoring the presence of people, and one in the front door monitoring the entrance of people in the house. The knowledge base includes the following definitions:

(1) $\texttt{Room} \sqcap \neg\exists\texttt{hasOccupant} \sqsubseteq \texttt{EmptyRoom}$
(2) $\texttt{EmptyRoom} \equiv \texttt{Room} \sqcap \neg\texttt{OccupiedRoom}$
(3) $\texttt{Room(x)} \wedge \texttt{EmptyHome(y)} \wedge$
$\quad\texttt{isInside(x,y)} \rightarrow \texttt{EmptyRoom(x)}$
(4) $\neg\texttt{EmptyRoom(x)} \rightarrow \texttt{OccupiedRoom(x)}$

In the example, the front door sensor asserts the entrance of one person, yet none of the room sensors succeeds to communicate subsequently that a person is present. Due to OWL's open world semantics, rule (4) evaluates to true for all rooms, as it cannot be proved that they are empty. As a result the system ends up inferring that there is at least one person present in each room.

4.4. Hybrid Reasoning with Statistical Techniques

Ontological reasoning has been mostly applied to inferring higher-level information, however it has limited ability to process low-level sensor data like acceleration, biometrics or sound data. These sensor data are often processed by a machine learning technique and derived to a high-level concept like "stand", "run", or "talk". Hence ontological reasoning is often integrated or paired with statistics-based techniques. In addition, the statistics-based technique can help to reduce performance bottleneck, as dealing with a large amount of uncertain, streaming data has been considered as the main constrain of ontological reasoning.

Not only can ontological reasoning benefit from statistical reasoning, but also the ontological knowledge can be used to guide and constrain the learning or inference process. Riboni et al. [110] specifies the knowledge between the domain concepts and the activities; e.g., BrushTeeth must happen in a room with a sink. They use a statistical technique to infer the current possible activities as {(BrushTeeth 0.6), (Reading, 0.5)}. If the room derived from the current sensor data does not have a sink, then the inference result is Reading, rather than BrushTeeth that has the higher probability. This demonstrates the earlier attempt to use predefined knowledge to help select the inference result more accurately.

Ye et al. [155] propose a general ontological model for representing human activities and domain concepts in a smart home environment including objects, locations and sensors. The ontological model is built on top of the standard knowledge base (e.g., WordNet), and has been demonstrated generic enough to be shared and reused in different home settings. The knowledge is deeply integrated in semantic similarity measure, clustering, and string alignment techniques to achieve unsupervised learning. This is a good example of how to embed basic knowledge in statistical reasoning but avoid over-engineering.

4.5. Analysis

According to the above discussion, two critical observations can be made. First, the combination of ontologies with rules is a key prerequisite for effectively meeting the expressiveness requirements when modelling and reasoning about context in the pervasive domain. However, hybrid reasoning schemes that either only lead to poor interaction between the two components or that fail to take into account the particularities of the co-existence of closed and open world semantics, may easily lead to incorrect inferences and an overall undesirable behaviour. Secondly, the combination of open- and closed-world reasoning is desirable when reasoning about context in the pervasive domain. The open-world semantics are closely related to the ability of reasoning over incomplete knowledge, while closed-world semantics is needed in order to reason over common conjectures about negative knowledge, without having to explicitly state such knowledge.

It is worth noting that ontology-based pervasive applications are not the only research domain where there is a need to reason, keeping certain parts of the world open while closing others. Similar challenges are confronted in the semantic understanding of visual content [36], and in general in applications that require intensional reasoning (e.g. natural language processing). Furthermore, such seamless integration of open and closed world semantics has been recognised as a highly challenging yet desirable capability in the Semantic Web too, resulting in a number of promising proposals as discussed in section 2.4. Their practical impact within the pervasive domain remains subject to future investigation.

Hybrid ontological and statistical reasoning is a must so that ontological techniques are able to deal with low-level signal processing and achieve real-time performance. The advantages of using ontological knowledge in statistical reasoning have also been witnessed in recent works; that is, guiding and constraining the reasoning to reduce the reliance on the training data.

5. Streaming Sensor Data Modelling, Querying, and Reasoning

As discussed so far, SW technologies provide rich ontology languages and powerful reasoning and querying mechanisms that meet the foundational requirements of typical pervasive systems. However, they normally deal with static data. The need to also handle real-time streaming data has been identified within several pervasive computing applications; characteristic examples include mobile telecommunications [82], public health risk monitoring applications (discussed in [39]) and traffic monitoring [62]. Temporal data processing and reasoning have been well studied in Database, Artificial Intelligence, and Formal Methods [49], and here we focus on the joint work with ontologies in the application area of pervasive computing.

5.1. Temporal Data modelling and Indexing

Temporal extensions to RDF, all notionally based on expanding the triple model to a quad model, have been explored in the literature. Lilis et al. [81] introduce *Multidimensional RDF*, an RDF extension designed to express the temporal semantics of a collection of cultural artefacts. Time is used as a contextual specifier to control whether or not an RDF triple should be considered to be present in a graph at a given time point or interval. Gutierrez et al. [59] provide the semantics for temporal RDF graphs, introducing the notion of a temporal triple, an RDF triple with a temporal label, and a temporal graph made from a set of temporal triples. The authors present the concepts of graph slices and graph snapshots, which allow for the description of the collection of triples that hold during or at a given interval or instant. Furthermore, using the notion of temporally-extended RDF, Pugliese et al. [106] introduce the *tGRIN* index structure that builds an index for storing temporal RDF in a relational database based on temporal as well as the structural closeness of triples, while Tappolet et al. [137] present a method for building a meta-index for the validity of named graphs based on Elmasari et al.'s *Time Index* [45].

5.2. Temporal Data Querying

Many extensions to RDF query languages have been proposed for working with temporal and streaming data; this is crucial to the process of querying for and temporally correlating information from different sensors as part of the query process. One of the earlier attempts involved attaching the temporal extent to objects and then apply a syntactic extension to SPARQL [85, 137]. O'Connor et al. developed a lightweight temporal model to encode data based on the valid-time dimension, upon which they develop extensions to their SWRL-based OWL query language *SQWRL* [96]. Their query library supports Allen's relations and provides functions for grouping query results such that the filters *first*, *first-n*, *last*, *last-n*, and *nth* can be applied within a query.

Beyond the above approaches on one-time-only processing, a number of SPARQL extensions have been proposed towards a framework where queries are evaluated continuously against new data being produced, with query results updated as new matches are discovered. The work mainly is to extend the SPARQL grammar to support streaming data representation and continuous queries evaluation via a sliding time window. The representative examples include *Streaming SPARQL* [18], *C-SPARQL* [9], and *CQELS* [78].

5.3. Temporal Data Reasoning

Moving beyond temporal extensions, the SW community has investigated hybrid frameworks that combine ontologies with temporally-aware formalisms. For example, the ambient computing framework proposed by [102] takes advantage of the inherent temporal reasoning capabilities of *Event Calculus* [77]. Rule-based reasoning is used on top of context modelling ontologies, to infer complex contexts from raw context data. In parallel, causality reasoning is employed to reason over pre-conditions and effects of actions and events, based on the Event Calculus theory. Compared to plain rule-based approaches, a key advantage is the inherent notion of time in Event Calculus that allows to establish a linear time ordering and, hence, infer in which intervals certain conclusions hold, while re-evaluating event patterns as time progresses. Batsakis et al. [10] propose an approach to support both qualitative and quantitative temporal reasoning. Here the temporal information is represented in `4D-fluents`, allowing for representing quantitative temporal information with specific temporal instants or intervals. Reasoning on quantitative temporal information is supported; for example, if interval A contains interval B and point C is into interval B we can infer using the point based representation that C is into interval A.

The situation awareness architecture presented in [5] is another example, where different formalisms and reasoners have been combined to enable inference on data that change over time. Sensor observations are first aggregated by means of `if-then` rules, and subsequently fed to the semantic interpretation layer, where a DL reasoner is used for situation assessment. The SCEP frameworks described in section 5.3.1, coupling the real-time reasoning capabilities of CEP engines with rich ontological semantics, are yet another example of investigations towards the efficient handling of temporal semantics.

A recent approach that captures reasoning with non-static data is *Stream Reasoning* [37], an attempt to combine data stream and reasoning technologies towards a solution for real-time reasoning over rapidly changing information. Stream Reasoning is defined as *'logical reasoning in real time on gigantic and inevitably noisy data streams in order to support the decision process of extremely large numbers of concurrent users'* [135]. In [39] a number of issues that need to be addressed in stream reasoning systems have been identified. These vary from theoretical aspects such as formal models, sound and complete reasoning mechanisms and algorithms to adequately address the stream reasoning-specific requirements, to more technical issues such as wrapping solutions for heterogeneous formats of dynamic data, solutions to the problems of noisy and uncertain data and parallelisation and distribution of various tasks to different units.

Theoretical investigations have led to a number of proposals towards stream reasoning languages. *Construction Description Logic* ($c\mathcal{ALC}$) is introduced to serve as a semantic type system and knowledge representation formalism for data streams [86]. $c\mathcal{ALC}$ is based on DLs, but its semantics are refined to a constructive notion of truth that captures the uncertainty aspects inherent with data streams. In [62], *DyKnow*, a stream-based knowledge processing middleware is introduced, which supports incremental reasoning with streams using *Metrical Temporal Logic* [99] as the underlying logical language. In [47], *LarKC* is introduced, a platform for Web-scale reasoning; further implementation attempts towards LarKC have led to the definition of its underlying language *L2* [50], a lightweight language based on the RDFS vocabulary and a limited subset of OWL. For more temporal description logics, we refer to the survey [83].

5.3.1. Semantic Complex Event Processing

State of the art *Complex Event Processing* (*CEP*) frameworks [118] are able to efficiently recognise complex events based on predefined event-patterns, event-hierarchies or other event relationships (e.g. temporal). CEP engines are capable of binding to input streams of real-time structured events generated either directly by sensors or after low-level feature processing. The key feature is the support of temporal relationships and aggregation operators that enable the identification of complex correlations among the generated events.

Semantic Complex Event Processing (*SCEP*) constitutes an effort to improve CEP's results by incorporating ontologies into the process of complex event detection [88]. An abstract SCEP architecture is presented in Figure 3. Ontologies are used in this context as common vocabularies for representing knowledge relevant to events. Low-level events are semantically associated with high-level domain concepts of background ontological knowledge, improving the quality of event and activity recognition using contextual information. Existing CEP engines or SPARQL temporal extensions can be used to process streams of events and uncover temporal correlations.

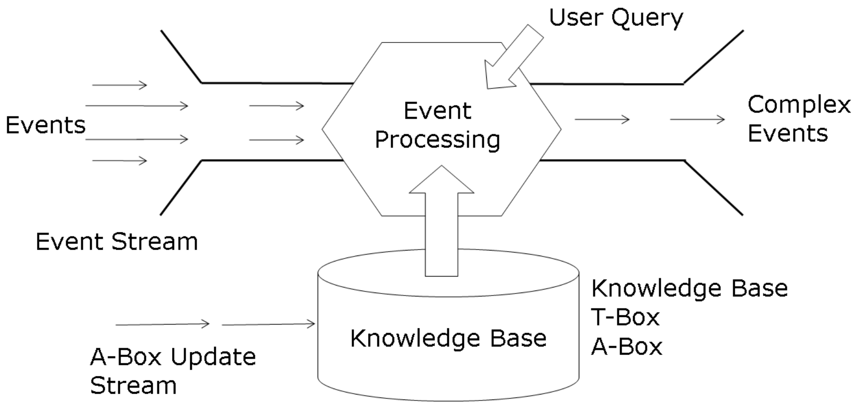


Figure 3: An abstract SCEP architecture (source: [101])

A commercial CEP engine (*Coral8*[11]) is combined with ontology-based reasoning [138]. The platform aims at the semantic configuration of the CEP engine using domain ontologies about events, sensors, environments, etc. The event ontology extends SSN (see section 3.1) and it is used as a repository of complex event definitions that users define. Complex events are described in terms of atomic events and the alerts that should be triggered upon detection. These events are transformed into CEP streams and queries for the configuration of the CEP engine.

Teymourian and Paschke present a framework for semantic rule-based complex events processing [140]. The ontology-based representation of event data is performed via a mapping of the attribute-value pairs to a set of RDF triples (RDF graph) that can be used as event instances. In this way, events are represented in terms of URIs and they can be further interlinked with other ontology-based domain knowledge. More complex events can be retrieved by executing SPARQL queries that match complex graph patterns. The reaction rule language of *Prova*[12] is used to implement temporal reasoning operators and to perform reasoning on the ontological knowledge.

ETALIS [4] is a CEP engine implemented in Prolog that supports the definition and execution of EP-SPARQL queries, as well as the use of background knowledge in a form of RDF ontologies. EP-SPARQL

[11] `http://www.aleri.com/products/aleri-cep/coral8-engine`

[12] `http://http://prova.ws/`

is used in ETALIS as the underlying query language for detecting events within a stream of RDF triples (low-level events). The queries are compiled into event-driven backward-chaining rules that can be mixed with other background knowledge.

5.4. Analysis

Table 5: Different approaches in modelling, querying, and reasoning on temporal data

Functionality		Techniques
Modelling	Use temporal dimension as contextual information indicating the lifetime of an RDF triple in a graph	[81]
	Organise temporal triples in a graph showing progression	[59]
Querying	Extend SPARQL Syntax for querying temporal dimension	[137],[85], [105]
	Support temporal relationships	SQWRL [96, 97]
	Extend SPARQL algebra to explore temporal semantics	Streaming SPARQL [18], C-SPARQL [8, 9], CQELS [78]
Reasoning	Combine with temporally-aware formalism (e.g. Event Calculus) to reason on static temporal data	*SPARQL-ST* [5, 77]
	Combine with formal temporal logic (e.g. Construction Description Logic and Metrical Temporal Logic) to reason on streaming temporal data	[86], [99]
	(Semantic) Complex Event Processing	[88] [138] [140] [3]

Given the general-purpose nature of the RDF model, it is no surprise that SPARQL is correspondingly flexible in its querying capability. However, when considering the particular goal of supporting pervasive computing, such a general purpose language has its limitations, i.e. the often verbose nature of queries to support common idioms, such as querying for the spatial or temporal semantics of data. The research into SPARQL extensions outlined here, although particularly focused on temporal modelling, shows a path according to which query languages may be adapted or extended to target specific needs of the pervasive domain. However, in order for this approach to succeed beyond the prototype stage, the widespread adoption of modelling standards for said domains is required (e.g. representing spatial, temporal and uncertain data) that as of writing do not yet exist. Future research in this area is, therefore, tightly bound with the development, evolution and adoption of conceptual models. In terms of reasoning on streaming data, despite the aforementioned efforts there is still a gap between the research on advanced reasoning techniques [39]. Some first steps have been made towards the needs identified in [38] for innovation in foundational theories specific to Stream Reasoning, as well as Stream Reasoning Engineering, but further developments are yet to be addressed [9].

Additionally, this section introduced the notion of semantic complex event processing (SCEP), an effort to combine SW technologies with traditional complex event processing frameworks. The underlying motivation is to enrich the pattern-based complex event detection capabilities of SCEP with reasoning over ontology-based event models and background knowledge. In this way, complex events can be detected and interpreted based on their hierarchical or temporal relationships to other events, as well as on their correlations with other relevant concepts of the domain. Currently, the existing complex event ontologies do not scale well, especially with multiple parallel queries [74]. With certain optimised implementation mechanisms (e.g.

caching and indexing intermediate query results), some SCEP engines such as CQELS [79] can perform well with growing static data. However, there is still room for improvement regarding the performance and scalability of SCEP engines in continuously querying over streaming sensor data. Table 5 summarises the section.

6. Uncertainty Handling

Because components of a pervasive computing environment deal with the real world, they may face certain caveats: sensors in the field may report inaccurately due to hardware fault or because they come up against an unusual phenomenon; i.e. one for which they have not been designed. Moreover, data are inherently imperfect and inaccuracies may easily arise, due to erroneous or missing sensor readings. Furthermore, when data comes from multiple sources and modalities, ambiguities and conflicts may arise. Under these circumstances, modelling and reasoning need to provide the means to cope with such imperfections and allow to detect possible errors, handle gracefully missing values, derive plausible conclusions, and assess the validity of the retrieved sensor data. Since these issues must be taken into consideration when dealing with pervasive systems, it should be possible to describe the concepts of accuracy, uncertainty, and provenance with respect to sensed data and represent them as part of the ontological description. With these descriptions in place, particular reasoning mechanisms on ontologies need to be designed to support efficient and precise reasoning on the data.

6.1. Imperfect Knowledge

Gaia, an early representative, tries to make sense of the imprecise and conflicting uncertainty inherent in dealing with real-world data [107]. An uncertainty model is developed based on a predicate-based representation of contexts and associated confidence values. To reason about uncertainty, Gaia employs probabilistic logic, fuzzy logic, and Bayesian networks, each of which offers certain advantages under different circumstances. The networks are trained with real data so as to get more accurate probability distributions for their event nodes.

This type of approach uses ontologies syntactically as a vocabulary for exchanging knowledge specified in a probabilistic model. Responding to the need of modelling imperfect knowledge in the Semantic Web, much research has been devoted to extending existing formalisms and reasoning services for handling uncertain and vague information. Representative examples include fuzzy extensions of DLs [132], OWL [17] and SWRL [151], and probabilistic extensions such as PR-OWL [24] and BayesOWL [40]. For an extensive overview the reader is referred to [133]. Further relevant proposals include the pattern-based approach for representing and reasoning with fuzzy knowledge [145], and the generic, formalised approach for managing uncertainty proposed by Helaoui et al. [63] use log-linear DLs to develop a probabilistic ontological framework to hierarchically recognise multi-user complex activities. Recent works have started exploring the applicability of such initiatives in the domain of pervasive applications; an example is the approach presented in [114], where fuzzy ontologies are applied to human activity recognition to deal with sensor unreliability and activity specification imprecision.

6.2. Missing Data

Missing data is another source of uncertainty when reasoning about context: a missed (or inaccurate) detection of low-level context information may easily lead to irrecoverable failures in the inference of higher-level context abstractions. One possible solution is to model the interpretation of perceptual data as inference to the best explanation using abductive reasoning [103, 122]. Romero et al. [53] investigate this idea in the context of an ontology-based surveillance application. A set of ontologies are used to capture context at increasing levels of abstractions, including tracking knowledge, scene objects and activities. Once the low-level context acquired from visual sensors is translated into ABox assertions, abductive rules are applied to derive missing facts and trigger the derivation of higher-level context descriptions. No information is provided about the computational framework used to implement the abductive reasoning, or about the preference criteria used to select explanations. As abduction is acknowledged as a mode of reasoning that is inherent in various tasks, much research has been devoted to understanding it.

Bikakis et al. [14] propose a formal model based on defeasible logic to support reasoning with imperfect context in ambient computing environments. Extending the Multi-Context Systems model with non-monotonic features, the proposed framework supports reasoning in cases of missing context knowledge. Potential inconsistencies are resolved by means of: (a) an argumentation framework that exploits context, and, (b) preference information that expresses confidence on the contexts considered. The propositional representation of context knowledge may not allow a direct integration with ontology-based context reasoning frameworks; yet possibilities for interesting hybrid architectures emerge where contextual assertions can be selectively translated into equivalent grounded formulas.

6.3. Analysis

At present, most ontology-based models in the pervasive computing community are still at the stage of using semantic annotations to tag different quality measures. One of the future directions in dealing with uncertainty is to use the ontologies that are tightly integrated with probabilistic or fuzzy reasoning. Abductive reasoning is worth further investigation, as it can help detect errors (e.g. missing or inaccurate data) that can be used as a feedback to re-tune the system. Also performance is still the main obstacle of ontological uncertainty reasoning, while the possible solution would be to combine data-driven techniques with fuzzy ontologies [114].

Table 6: Different approaches in modelling and reasoning on uncertain data

Functionality	**Techniques**
Representing uncertain data with probabilities	Gaia [107]
Reasoning on uncertain data	PROWL [24, 34], BayesOWL [40], [29, 41, 114, 145] based on Fuzzy logic
Dealing with missing data	[103, 122] based on abductive reasoning
Resolving inconsistent data	[14] based on defeasible logic

7. Challenges and Open Areas

The previous sections have discussed the key requirements in pervasive computing and have analysed the benefits and potentials of using SW technologies, along with future research inquiries and directions. The discussion has been structured along the four key requirements, as identified in the Introduction: (a) conceptual modelling (ranging from raw sensor data to higher-level context and event abstractions), (b) reasoning, (c) temporal data modelling, querying and reasoning, and, (d) handling uncertainty. Besides outlining the strengths and weaknesses of the proposed approaches, the sections also presents the relevant results within the SW community, which have not yet been explored in the context of pervasive systems, sketching possible directions for further investigations.

Table 7 summarises the main observations. The use of SW technologies induces a number of straightforward benefits, as a direct result of the advocated explicit semantics and well-defined reasoning services. Issues open to further investigation are to a large extent inherited by challenges that remain open in SW research, too. Scalability and performance are crucial, as the need to draw inferences from millions or billions of pieces of information in real-time is not restricted in the domain of pervasive systems alone. With this concern in mind, d'Aquin et al. [35] conduct a series of experiments on assessing the performance of existing semantic tools including Jena, Sesame and Mulgara on resource-constrained devices (e.g., a netbook with 900 MHz CPU and 512 MB RAM). The evaluation metrics contain not only the size of data and the response time, but also the device specifications such as the memory and disk space, and the nature of data such as the distribution of entities in classes, properties and individuals. The results show that these tools are able to cope reasonably well with small-scale ontologies on such devices. However, when faced with

Table 7: Overview of Semantic Web technologies use in pervasive applications with respect to key requirements.

Requirement	**SW-empowered approach**	**Added value**	**Further research inquiries**
Conceptual modelling	ontologies for describing sensor data, who/where/what/when information, events; rules for higher relational expressiveness	facilitating knowledge sharing, reuse and exchange; rich expressiveness for capturing complex semantic relations	upper ontologies; comparative assessment of existing ontologies
Reasoning	DLs reasoning services; rule-based reasoning; reasoning with ontologies and rules	inferring implicit knowledge; consistency checking	scalability; combining open- and closed-world reasoning; reasoning under inconsistency
Temporal semantics	OWL-Time, Semantic CEP, RDF/SPARQL extensions, hybrid frameworks (e.g. ontologies and Event Calculus)	SPARQL and its extensions to support temporal and stream queries	highly flexible and general purpose query language supporting the inspection of static and streaming data; further development of domain specific constructs to simplify querying; highly dependent on standardisation and adoption of conceptual models; temporal/real-time reasoning; hybrid reasoning frameworks
Uncertainty handling	extensions for fuzzy / probabilistic semantics (e.g. Fuzzy DLs, PR-OWL); non-standard inference (e.g. abduction in DLs ABox)	reasoning over imprecise/vague knowledge; reasoning to hypotheses	scalability; seamless integration of uncertainty; unified handling of different types of uncertainty

millions or billions of triples as mentioned above, we consider the future work should focus on designing a fully distributed approach for coordinating and integrating the reasoning capabilities on low-end devices (especially sensors and mobile devices) to enhance performance.

The seamless integration of open- and closed-world reasoning is another highly desirable feature and subject of active research, and the same holds for investigations into the practical management of imperfect (uncertain, vague, missing, noisy) knowledge. Last but not least, throughout the study of SW-based pervasive frameworks, it is evident that relevant insights often permeate only partially, and in a fragmented manner, the borders between the two research communities. In the following, we will discuss open research issues relevant to *temporal features*, *dynamism*, *provenance*, and *programming* that need also to be investigated towards the seamless integration of SW technologies and pervasive computing.

7.1. Temporal Features

The role of temporal information in the modelling of sensed data is often simplified in the design of data models, software and application APIs. Typically, sensor values are recorded in conjunction with a *time-stamp*. The use of timestamps may provide a basis in addressing the need to establish temporal relations to correlate information from different sensors for querying. However, despite its universality as a modelling concept, the use of timestamps in isolation implicitly restricts a data model to representing only the current state. That is, a timestamp has an implicit dual meaning: it is both the time at which a statement was asserted in the model and the time at which the statement should be interpreted as being *true*. This distinction may often seem unimportant, however, the ability to separately annotate data with *temporal extent*, i.e. a time interval, caters explicitly for the latter case. Furthermore, the use of both an update time-stamp and temporal extent in concert provides a useful facility for modelling both historical and predictive state. This is often useful; for example, summarising a high volume of sensor readings over a period of time is preferable to removing it entirely in the case where data will later be analysed off-line. modelling time as an orthogonal concern allows such cases to be handled without requiring special indicators in a data model to indicate that information has been treated.

Despite strong vocabulary support for representing temporal features, as discussed in section 3.2.1, the triple-based nature of the RDF model is ill-suited to the representation of data's temporal properties, where the requirement is to annotate sets of triples. At least three possible strategies for overcoming this limitation are available to data modellers: using RDF's reification vocabulary, externally generating identifiers for statements, or using a non-standard extension to the RDF model. To utilise RDF's reification vocabulary, each triple is expanded as a '*reification quad*'[13], with the resulting statement identifier associated with temporal information. Generating a statement identifier externally follows similarly, but avoids the introduction of redundant information into the model at the expense of requiring non-standard tooling to generate and resolve such identifiers.

Section 5.3 discussed temporal extensions to SW technologies and reasoning mechanisms on temporal and streaming data. However, there is still a gap between the currently available SW technologies and the need for native temporal data modelling and reasoning. Relevant prototype suggestions exist, but considerable effort is required before reaching standardised solutions that will in turn ensure the required tool support for efficient reasoning over temporal data and management of temporal queries. How to correlate sensor data collected from various sources in terms of their temporal relations to support queries is also worthy of investigation. Towards this quest, the combination of different formalisms and reasoners within hybrid frameworks can lead to extended representational and reasoning capabilities. On the other hand, the seamless and scalable integration of the heterogeneous modules inevitably poses new challenges.

7.2. Data Dynamics

Distinct from the need to model the temporal features of dynamic data are the challenges related to storing, reasoning over, and accessing large volumes of highly-dynamic, distributed information. As the volume

[13] The term 'reification quad' refers to a set of 4 triples that associate an identifier with the type `rdf:Statement`, and three properties, `rdf:subject`, `rdf:predicate`, and `rdf:object` with values corresponding to the original triple.

of sensor data increases, it becomes infeasible to store a permanent record of generated states. A typical solution might be to discard the oldest data from memory when the maximum capacity is reached. However, the oldest data stored is not necessarily the least useful. An alternative solution is to modify the data model for capturing properties describing data dynamics. This may lead to defining policies for prioritising the removal of data that is asserted frequently but has a slow rate of change (e.g. ambient temperature) over data that is frequently asserted but changes rapidly (e.g. a user's coordinate location) or pseudo-static data that is infrequently asserted (e.g. building layouts) [130].

Data dynamics also play a role in determining the types of reasoning strategies that should be adopted. There is a need to track derivations in order to determine when an inference no longer holds due to modifications to (or invalidation of) the data upon which it is predicated. Structuring the data model in such a way that data is temporally qualified (and inferences likewise) forms part of a solution to this issue, and requires incorporating appropriate temporal semantics with existing reasoning technologies.

The process of reasoning over a large data model can be a performance bottleneck. However, knowledge about the dynamics of data can inform an appropriate reasoning strategy. For example, one might choose to reason on static data as it is generated and at the same time perform reasoning on highly dynamic data only when an application query is executed. In this way reasoning would be restricted to the smallest amount of volatile data that will produce a correct answer. Based solely on the temporal properties of data, it may be possible to devise a general scheme to partition, reason on, and integrate data over several stages, so as to optimise reasoning.

7.3. Provenance

Many of the issues above can be subsumed under the notion of *provenance*, i.e. capturing, representing and manipulating knowledge relevant to data creation, ownership, transformation and other 'life-cycle' issues. Provenance appears in many guises. The Dublin Core vocabulary is perhaps the most well-known paradigm, providing terms for asserting authorship and other property rights over digital objects. More recently, a task group of the World Wide Web Consortium has been standardising the *Open Provenance Model* [87] for asserting more general provenance metadata. Sensor-driven systems are more directly affected by data provenance than programs in many other domains. A sensor system must make decisions using input data that is known to be inherently noisy, imprecise, inaccurate, untimely and infrequent. An immediate consequence is that sensor-driven systems cannot be directly connected (or respond directly) to their input data streams. Stating it differently, individual data elements are *evidence of fact* rather than being *facts themselves*, and must be fused with other data (from the same or different sources) to build a consensus of the state of the environment being sensed. Chowdhurry et al. [28] propose a context composition graph to represent the hierarchical process by which high-level contextual inferences are made by composing low-level sensor-generated data samples. It also allows to track the temporal history of contextual states along the hierarchy.

While sensor fusion is commonplace in engineering, many of the approaches make strong assumptions about the nature of the data streams; for example, they assume that data comes from homogeneous sensors looking at the same phenomenon and with well-known sampling frequencies. However, such assumptions do not generalise well to semantically-enriched systems with highly heterogeneous data sources processed using symbolic reasoning. Note that this issue also applies to many other aspects other than provenance, such as querying and reasoning. Conversely, homogeneous data can be easily handled by reasoners, allowing the semantic technology to encompass many sensor fusion tasks. Moreover, reasoning over sensor data is affected by a number of provenance-related factors. A trivial example is that the reliability of a temperature reading retrieved from data that is several hours old may be assumed to be less than that inferred from data collected within the previous minute. Sensor types (and even individual sensors) give rise to data with given provenance in terms of the observation frequency, precision etc., which can be captured generically or via special-purpose markup languages (see section 3.1). The provenance here is *associated* with the sensor but is *attached to* the data produced by the sensor.

More generally, a system may be interested in the entire lifecycle of a data stream, including the transformations that have been applied to it. Suppose that a stream of temperature values is received, which will be used in some decision process. It may matter whether those values are 'raw' (and subject to raw sensor noise), they have been processed to remove outliers, or they have been smoothed using an *a priori* or learned smoothing function. Thus, it is vital to manipulate and maintain provenance along the data pathway. For many systems the statistical properties of a data stream are as important as the data itself. Extensive in-system processing of data is not necessarily a problem, *if* it is clear to the end-user that such processing has occurred and to what effect.

Provenance is increasingly recognised as a crucial element of the Semantic Web, as it enables inferences to be drawn conditional to whether information should be trusted and how it should be reused and integrated with other diverse information sources. The lack of a standard model for capturing, interchanging and reasoning over provenance metadata is a significant impediment to realising applications where the trustworthiness and the quality of the statements is an issue. The *Provenance Working Group*[14] has recently concluded the standardisation of an interchange core language (*PROV Ontology*[15]) for publishing and accessing provenance metadata, drawing on existing vocabularies and ontologies. In parallel, recent studies have proposed approaches that enable the handling of provenance in open and collaborative environments [98, 141]. Also Riboni and Bettini have proposed the first provenance framework for representing and reasoning on context provenance with a focus on uncertainty and temporal aspects in ambient intelligent systems [113].

7.4. Programming

Lastly we come to the practicalities of programming. Semantic structures like RDF allow rich data encodings. Coupled with OWL, SPARQL, ontological and other reasoners, complex queries can be answered by traversing the knowledge graph. In this context, the emergence of standard, re-usable reasoners significantly simplifies the use of semantic technologies within programs.

On the practical side, however, the integration of these tools remains superficial. From a programming perspective, a knowledge graph is simply a collection of edges labeled with strings and URIs, possibly with some additional XML Schema typing at the endpoints, and with some structure on the relationships provided by the accompanying ontologies (if any). This often leads programmers to attempt to syntactically and structurally access knowledge graphs, using well-known and established parsing tools, but ignoring the underlying semantics. In this quest, developers are typically forced to manually provide any needed additional mechanisms without the assistance of compilers, type systems or other tooling, which constitute common practice while building complex software systems outside the Semantic Web [128]. In an effort to simplify the integration of SW technologies into existing software architectures and languages, various APIs for working with RDF/OWL ontologies have been developed, such as *Jena*[16], *OWL API* [66], *dotNetRDF*[17], *RDFReactor* [148] and *Sesame*[18]. Moreover, Janowicz et al. [70] developed a promising approach based on the Linked Data model and a RESTful proxy to publish sensor data on the Web (see section 3.1.2).

Furthermore, the use of semantic technologies requires considerable familiarity with XML tools that obscure rather than illuminate the underlying information being encoded. Data encoded in this manner is not held in this form in memory and must be translated for storage and exchange, which is a non-trivial process. State of the art ontology editors, such as *Protégé* [95], *TopBraid Composer*[19] and *NeOn Toolkit*[20] offer comprehensive support for developing and validating RDF/OWL ontologies. Moreover, scalable and query efficient RDF repositories, such as *OWLIM* [15], *AllegroGraph*[21] and *OpenLink Virtuoso* [46], as well as database-to-RDF mapping tools such as *D2RQ* [44], enable data to be exposed and shared on the Web according to the principles of Linked Data.

Entities represented within a knowledge graph may be classified by the ontology according to the information related to them; e.g. classifying a person as an employee given the presence of an employee number.

[14] `http://wiki.knoesis.org/index.php/Provenir_Ontology`
[15] `http://www.w3.org/TR/prov-o/`
[16] `http://jena.apache.org/`
[17] `http://www.dotnetrdf.org/`
[18] `http://www.openrdf.org/`
[19] `http://www.topquadrant.com/tools/modeling-topbraid-composer-standard-edition/`
[20] `http://neon-toolkit.org`
[21] `http://www.franz.com/agraph/allegrograph/`

This is roughly the same as a programming language type system, with the difference that an entity's class may change unpredictably through other agents' actions on the knowledge graph, and it can be hard to predict precisely *which* such changes will have such an effect. This destabilises a program's view of the knowledge in the graph.

The use of a reasoner sits outside the programming language, in the same way that SQL does when accessing a database. A SPARQL query is a string that returns arrays of other strings and is not type-checked or manipulated using dedicated programming constructs. This complicates the formation and checking of complex queries, again due to a lack of supporting tooling. However, some of the aforementioned ontology editors do provide limited support for advanced query formulation, e.g. type-checking.

Finally, the Semantic Web may pose steep learning and commitment curves, particularly for practitioners coming from other research areas. In order to perform even simple tasks, developers must master a wide range of potentially unfamiliar technologies. Moreover, an organisation must commit to these technologies and the respective costs ahead of time. This raises the risk that a system may not generate the expected benefits while incurring a substantial up-front cost and thus, raises a significant barrier to deployment.

Consequently, as with any technology, the decision to adopt SW technologies, does not come without a price, and one needs to ascertain sufficient value on its advantages – open, standards-based representation, easy exchange and integration – to make it worthwhile. It is undoubtedly attractive to be able to define a structure for knowledge that exactly matches a chosen sub-domain, to describe the richness of this structure, and to have it integrate efficiently with other such descriptions of complementary sub-domains defined independently – and to be able to share all this knowledge with anyone on the Web. But this flexibility comes with a cost and (often) no apparent immediate, high-value benefits. In many ways, these issues are an inevitable consequence of the differences in the application domains of semantic technology and programming languages. The former addresses the open, extensible, scalable, distributed mark-up of data. The latter addresses almost the opposite issues, focusing on close specification of algorithms and data structures to share common data and functionality.

Since these issues apply to *all* programming with semantic structures, sensor-driven systems add few, if any, specific points to this general discussion. Sensor-driven systems impose the necessity to embrace the noise and errors inherent in data streams, and to make decisions that propagate this uncertainty throughout the system. Mainstream programming languages do not provide structures for resolving these issues. Combining programming with uncertainty and with data that adhere to an ontological structure seems to be two new frontiers in systems design, driven by semantically-enriched sensor-driven systems.

Perhaps the chief impediment to such designs comes from the challenged nature of the platforms themselves. Sensor-driven systems place a significant degree of their functionality on devices with extremely constrained memory, computation and communication capabilities, which are not straightforwardly subject to Moore's-law-driven improvements in performance. A naïve deployment of SW technologies to such platforms is doomed to failure, but there seems to be no *a priori* reason why versions optimised for restricted domains might not be possible, and might not inter-operate easily with the wider universe of web-enabled components.

8. Discussion and Concluding Remarks

This article reviewed the landscape of the applications of SW technologies in the domain of pervasive, adaptive and sensor-driven systems. Many of the features underpinning the Semantic Web, and particularly the ability to formally capture and reason over rich, semantic interconnections among data items in an open and extensible way, are well-suited to pervasive computing. On the other hand, a number of other important aspects, such as the lack of native support for representing and reasoning over temporal or imperfect knowledge, remain highly under-articulated.

In terms of modelling and reasoning about context, the key benefits of SW technologies are straightforward. As mentioned previously, ontologies have been proposed for describing the four *Ws* (*when, where, what, who*) that characterise contextual knowledge, complex events and sensor data. This facilitates the unambiguous sharing and understanding of knowledge across heterogeneous and distributed platforms, devices and services. The modelling, reasoning, and uncertainty management issues among the five challenges identified by Corcho and Carcía-Castro [33] have been resolved to different degrees. Acknowledging the diverse views that different ontologies inevitably serve, what still remains unaddressed is the definition of commonly-agreed (possibly upper) ontologies that would enable the standardised description of pertinent context aspects; the conceptual overlaps between the proposed ontologies further underline this need. Moreover, since not all of the provided modelling capabilities have been directly applied in pervasive applications (e.g. modelling of composite events and their dependencies that many of the event ontologies support), it is time to systematically assess their applicability, the potential for their combined use or (partial) alignment, as well as aspects that require a more elaborate coverage.

In parallel, the deployment of SW technologies has demonstrated the intrinsic relation between automated reasoning and the high-level interpretation of context data, where the integration of structured domain knowledge is a prerequisite. Besides useful insights on the need to combine ontologies with rules, the reviewed literature sketches a number of desirable, yet highly challenging questions. These include the need to provide native support for representing and reasoning over temporal knowledge, incorporating provenance into reasoning, and managing uncertainty. Underlying all the aforementioned research directions, an indispensable requirement is computational efficiency. Ensuring reasoning efficiency is crucial, as the need to draw real-time inferences from millions or billions of information items in an open pervasive environment already challenges existing reasoning engines. And one should note that only deductive inference over crisp, consistent knowledge bases is considered.

Subject to further investigation is also the overall integration of SW technologies within programming frameworks suitable for software engineering in-the-large. The Semantic Web at present is essentially a collection of fragments lacking a whole. This is perhaps an inevitable consequence of an architecture designed for such a broad spectrum of application domains, but it nevertheless increases the risks and costs associated with applying the technologies to pervasive systems. This is especially the case when the target platforms for much of the functionality are challenged in terms of their capabilities in memory, computation and communications. However, it is important to remember that the SW is essentially a tool for modelling and exchange and not that much a tool for implementation: compact representations of information and reasoning that comply to the underlying meta-model of the SW seem eminently possible and deserve future exploration.

Summing up, the ability to exchange models and data, to reason openly, to capture an extending set of data and metadata, and to interact with other web-enabled elements, all encourages the view that building future pervasive and sensor-driven systems around these technologies would lead to significant improvements in interoperability and semantic clarity. The essential prerequisite is, though, that the disparate elements can be integrated into a framework appropriate for system developers. Such an integration would allow innovation to proceed more rapidly and soundly, bringing significantly closer the vision of pervasive computing for seamless, integrated pervasive and sensor systems.

9. Acknowledgements

This work has been supported by the EU FP7 projects *SAPERE: Self-aware Pervasive Service Ecosystems* under contract No. 256873 and *Dem@Care: Dementia Ambient Care – Multi-Sensing Monitoring for Intelligent Remote Management and Decision Support* under contract No. 288199.

References

[1] J. F. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23(2):123–154, 1984.

[2] N. Andrienko, G. Andrienko, N. Pelekis, and S. Spaccapietra. Basic concepts of movement data. In F. Giannotti and D. Pedreschi, editors, *Mobility, Data Mining and Privacy*, pages 15–38. Springer Berlin Heidelberg, 2008.

[3] D. Anicic, P. Fodor, S. Rudolph, and N. Stojanovic. Ep-sparql: a unified language for event processing and stream reasoning. In *Proceedings of the 20th international conference on World wide web*, WWW '11, pages 635–644, Hyderabad, India, 2011. ACM.

[4] D. Anicic, S. Rudolph, P. Fodor, and N. Stojanovic. Stream reasoning and complex event processing in etalis. *Semantic Web - Interoperability, Usability, Applicability*, 1:1–5, 2011.

[5] F. Baader, A. Bauer, P. Baumgartner, A. Cregan, A. Gabaldon, K. Ji, K. Lee, D. Rajaratnam, and R. Schwitter. A novel architecture for situation awareness systems. In *TABLEAUX*, pages 77–92, 2009.

[6] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

[7] F. Baader and U. Sattler. An overview of tableau algorithms for description logics. *Studia Logica*, 69(1):5–40, 2001.

[8] D. Barbieri, D. Braga, S. Ceri, E. Della Valle, and M. Grossniklaus. C-SPARQL: A Continuous Query Language for RDF Data Streams. *International Journal of Semantic Computing (IJSC)*, 4(1), 5 2010.

[9] D. Barbieri, D. Braga, S. Ceri, E. Della Valle, and M. Grossniklaus. Stream reasoning: Where we got so far. In *Proceedings of the 4th workshop on new forms of reasoning for the Semantic Web: Scalable & dynamic*, pages 1–7, 2010.

[10] S. Batsakis, K. Stravoskoufos, and E. Petrakis. Temporal reasoning for supporting temporal queries in owl 2.0. In A. Kanig, A. Dengel, K. Hinkelmann, K. Kise, R. Howlett, and L. Jain, editors, *Knowledge-Based and Intelligent Information and Engineering Systems*, volume 6881 of *Lecture Notes in Computer Science*, pages 558–567. Springer Berlin Heidelberg, 2011.

[11] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):34–43, May 2001.

[12] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, and D. Riboni. A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing*, 6(2):161–180, 2010.

[13] G. Biamino. Modeling social contexts for pervasive computing environments. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on*, pages 415–420, March 2011.

[14] A. Bikakis and G. Antoniou. Contextual argumentation in ambient intelligence. In *Proceedings of the 10th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR '09)*, pages 30–43, Potsdam, Germany, 2009. Springer-Verlag.

[15] B. Bishop, A. Kiryakov, D. Ognyanoff, I. Peikov, Z. Tashev, and R. Velkov. OWLIM: A family of scalable semantic repositories. *Semant. web*, 2(1):33–42, Jan. 2011.

[16] C. Bizer, T. Heath, and T. Berners-Lee. Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 5(3):1–22, MarMar 2009.

[17] F. Bobillo and U. Straccia. Fuzzy ontology representation using OWL 2. *Int. J. Approx. Reasoning*, 52(7):1073–1094, 2011.

[18] A. Bolles, M. Grawunder, and J. Jacobi. Streaming SPARQL extending SPARQL to process data streams. In *Proceedings of the 5th European semantic web conference on The semantic web: research and applications*, ESWC'08, pages 448–462, Berlin, Heidelberg, 2008. Springer-Verlag.

[19] S. Borgo, N. Guarino, and C. Masolo. A pointless theory of space based on strong connection and congruence. In L. C. Aiello, J. Doyle, and S. Shapiro, editors, *KR'96: Principles of Knowledge Representation and Reasoning*, pages 220–229. Morgan Kaufmann, San Francisco, California, 1996.

[20] B. Bouchard and A. Bouzouane. A key hole plan recognition model for alzheimer's patients: First results. *Applied Artificial Intelligence*, 22:1–34, 2007.

[21] A. Bröring, K. Janowicz, C. Stasch, and W. Kuhn. Semantic challenges for sensor plug and play. In *Proceedings of the 9th International Symposium on Web and Wireless Geographical Information Systems*, W2GIS '09, pages 72–86, Berlin, Heidelberg, 2009. Springer-Verlag.

[22] A. Bröring, P. Maué, K. Janowicz, D. Nüst, and C. Malewski. Semantically-Enabled Sensor Plug & Play for the Sensor Web. *Sensors*, pages 7568 – 7605, 2011.

[23] J.-P. Calbimonte, H. Jeung, O. Corcho, and K. Aberer. Semantic sensor data search in a large-scale federated sensor network. In *Proceeding of International Workshop on Semantic Sensor Networks (SSN)*, 2011.

[24] R. N. Carvalho, K. B. Laskey, and P. C. G. da Costa. PR-OWL 2.0 - Bridging the gap to OWL semantics. In *URSW*, pages 73–84, 2010.

[25] H. Chen, F. Perich, T. Finin, and A. Joshi. SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications. In *First Annual International Conference on Mobile and Ubiquitous Systems*, Boston, MA, USA, August 2004.

[26] L. Chen, C. Nugent, M. Mulvenna, D. Finlay, and X. Hong. Semantic smart homes: Towards knowledge rich assisted living environments. *Intelligent Patient Management*, 189:279–296, 2009.

[27] L. Chen, C. D. Nugent, and H. Wang. A knowledge-driven approach to activity recognition in smart homes. *Knowledge and Data Engineering, IEEE Transactions on*, 24(6):961–974, 2012.

[28] A. Chowdhury, B. Falchuk, and A. Misra. Medially: A provenance-aware remote health monitoring middleware. In *Pervasive Computing and Communications (PerCom), 2010 IEEE International Conference on*, pages 125–134, March 2010.

[29] A. Ciaramella, M. G. C. A. Cimino, F. Marcelloni, and U. Straccia. Combining fuzzy logic and semantic web to enable situation-awareness in service recommendation. In *Proceedings of the 21st international conference on Database and expert systems applications: Part I*, DEXA'10, pages 31–45, Berlin, Heidelberg, 2010. Springer-Verlag.

[30] M. Compton, P. Barnaghi, L. Bermudez, R. Garcia-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog, V. Huang, K. Janowicz, W. D. Kelsey, D. L. Phuoc, L. Lefort, M. Leggieri, H. Neuhaus, A. Nikolov, K. Page, A. Passant, A. Sheth, and K. Taylor. The SSN Ontology of the W3C Semantic Sensor Network Incubator Group. *Web Semantics: Science, Services and Agents on the World Wide Web*, 0(0), 2012.

[31] M. Compton, C. Henson, L. Lefort, H. Neuhaus, and A. Sheth. A survey of the semantic specification of sensors. In *Proceedings of the 2nd International Workshop on Semantic Sensor Networks (SSN09)*, pages 17–32, 2009.

[32] D. J. Cook, J. C. Augusto, and V. R. Jakkula. Ambient intelligence: Technologies, applications, and opportunities. *Pervasive and Mobile Computing*, 5(4):277–298, August 2009.

[33] O. Corcho and R. Garcia Castro. Five challenges for semantic sensor web. *Semantic Web - Interoperability, Usability, Applicability*, 1(1-2):121– 125, Dec. 2010.

[34] P. C. Costa, K. B. Laskey, and K. J. Laskey. PR-OWL: A bayesian ontology language for the semantic web. In P. C.

Costa, C. D'Amato, N. Fanizzi, K. B. Laskey, K. J. Laskey, T. Lukasiewicz, M. Nickles, and M. Pool, editors, *Uncertainty Reasoning for the Semantic Web I*, pages 88–107. Springer-Verlag, Berlin, Heidelberg, 2008.

[35] M. D'Aquin, A. Nikolov, and E. Motta. How much semantic data on small devices? In *Proceedings of the 17th international conference on Knowledge engineering and management by the masses*, EKAW'10, pages 565–575, Berlin, Heidelberg, 2010. Springer-Verlag.

[36] S. Dasiopoulou and I. Kompatsiaris. Trends and issues in description logics frameworks for image interpretation. In *Proceedings of the 6th Hellenic conference on Artificial Intelligence: theories, models and applications*, SETN'10, pages 61–70, Athens, Greece, 2010. Springer-Verlag.

[37] E. Della Valle, S. Ceri, D. Barbieri, D. Braga, and A. Campi. A first step towards stream reasoning. *Future Internet–FIS 2008*, pages 72–81, 2009.

[38] E. Della Valle, S. Ceri, D. Braga, I. Celino, D. Frensel, F. van Harmelen, and G. Unel. Research chapters in the area of stream reasoning. *SR2009*, 466, 2009.

[39] E. Della Valle, S. Ceri, F. van Harmelen, and D. Fensel. It's a streaming world! reasoning upon rapidly changing information. *Intelligent Systems, IEEE*, 24(6):83–89, 2009.

[40] Z. Ding, Y. Peng, R. Pan, Z. Ding, Y. Peng, and R. Pan. BayesOWL: Uncertainty modeling in semantic web ontologies. *Soft Computing in Ontologies and Semantic Web*, pages 3–29, 2006.

[41] R. Dividino, S. Sizov, S. Staab, and B. Schueler. Querying for provenance, trust, uncertainty and other meta knowledge in rdf. *Web Semant.*, 7:204–219, September 2009.

[42] S. Dobson. Leveraging the subtleties of location. In *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies*, sOc-EUSAI '05, pages 189–193, Grenoble, France, 2005. ACM.

[43] M. Doerr. The CIDOC conceptual reference module: an ontological approach to semantic interoperability of metadata. *AI Mag.*, 24(3):75–92, Sept. 2003.

[44] V. Eisenberg and Y. Kanza. D2rq/update: updating relational data via virtual rdf. In *Proceedings of the 21st international conference companion on World Wide Web*, WWW '12 Companion, pages 497–498, Lyon, France, 2012. ACM.

[45] R. Elmasri, G. T. J. Wuu, and Y.-J. Kim. The time index—an access structure for temporal data. In *Proceedings of the sixteenth international conference on Very large databases*, pages 1–12, Brisbane, Australia, 1990. Morgan Kaufmann Publishers Inc.

[46] O. Erling. Virtuoso, a Hybrid RDBMS/Graph Column Store. *IEEE Data Eng. Bull.*, 35(1):3–8, 2012.

[47] D. Fensel, F. van Harmelen, B. Andersson, P. Brennan, H. Cunningham, E. Della Valle, F. Fischer, Z. Huang, A. Kiryakov, T. Lee, et al. Towards larkc: a platform for web-scale reasoning. In *Semantic Computing, 2008 IEEE International Conference on*, pages 524–529. IEEE, 2008.

[48] L. Ferrari, A. Rosi, M. Mamei, and F. Zambonelli. Extracting urban patterns from location-based social networks. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Location-Based Social Networks*, LBSN '11, pages 9–16, New York, NY, USA, 2011. ACM.

[49] M. Fisher, A. Artale, Engineering, and P. S. R. Council. *Temporal Representation and Reasoning*. IEEE Computer Society Press, 2002.

[50] B. B. Florian Fischer, Gulay Unel and D. Fensel. Towards a Scalable, Pragmatic Knowledge Representation Language for the Web. 04 2009.

[51] A. Gangemi. DOLCE+DnS Ultralite (DUL) ontology. `http://www.loa.istc.cnr.it/ontologies/DUL.owl`, July 2012.

[52] A. Gangemi and P. Mika. Understanding the semantic web through descriptions and situations. In *Proceedings of the International Conference on Ontologies, Databases and Applications of SEmantics*, pages 689–706, 2003.

[53] J. Gómez-Romero, M. A. Patricio, J. García, and J. M. Molina. Ontology-based context representation and reasoning for object tracking and scene interpretation in video. *Expert Syst. Appl.*, 38(6):7494–7510, 2011.

[54] C. Goumopoulos, A. D. Kameas, and A. Cassells. An ontology-driven system architecture for precision agriculture applications. *IJMSO*, 4(1/2):72–84, 2009.

[55] B. C. Grau, I. Horrocks, B. Motik, B. Parsia, P. Patel-Schneider, and U. Sattler. OWL 2: The Next Step for OWL. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(4):309–322, October 2008.

[56] B. N. Grosof, I. Horrocks, R. Volz, and S. Decker. Description logic programs: combining logic programs with description logic. In *WWW*, pages 48–57, 2003.

[57] T. Gu, X. H. Wang, H. K. Pung, and D. Q. Zhang. An Ontology-based Context Model in Intelligent Environments. In *Proceedings of the Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2004)*, pages 270–275, January 2004.

[58] T. Gu, X. H. Wang, H. K. Pung, and D. Q. Zhang. An ontology-based context model in intelligent environments. In *Communication Networks and Distributed Systems Modeling and Simulation*, pages 270–275, 2004.

[59] C. Gutierrez, C. A. Hurtado, and A. Vaisman. Introducing time into rdf. *IEEE Transactions on Knowledge and Data Engineering*, 19(2):207–218, 2007.

[60] V. Haarslev and R. Möller. Racer: A Core Inference Engine for the Semantic Web. In *Proceedings of the 2nd International Workshop on Evaluation of Ontology-based Tools (EON2003), located at the 2nd International Semantic Web Conference, Sanibel Island, Florida, USA*, pages 27–36, 2003.

[61] J. Han, J.-G. Lee, H. Gonzalez, and X. Li. Mining massive rfid, trajectory, and traffic data sets. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, New York, NY, USA, 2008. ACM.

[62] F. Heintz, J. Kvarnström, and P. Doherty. Stream reasoning in dyknow: A knowledge processing middleware system. In *Proc. 1st Intl Workshop Stream Reasoning*, 2009.

[63] R. Helaoui, D. Riboni, and H. Stuckenschmidt. A probabilistic ontological framework for the recognition of multilevel human activities. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '13, pages 345–354, New York, NY, USA, 2013. ACM.

[64] M. Hepp. Goodrelations: An ontology for describing products and services offers on the web. In *Proceedings of the 16th international conference on Knowledge Engineering: Practice and Patterns*, EKAW '08, pages 329–346, Berlin, Heidelberg, 2008. Springer-Verlag.

[65] J. R. Hobbs and F. Pan. An ontology of time for the semantic web. *ACM Transactions on Asian Language Information Processing (TALIP)*, 3(1):66–85, 2004.

[66] M. Horridge and S. Bechhofer. The OWL API: A Java API for OWL ontologies. *Semant. web*, 2(1):11–21, Jan. 2011.

[67] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. Technical report, National Research Council of Canada, Network Inference, and Stanford University, May 2004.

[68] S. Hossein, S. Hedal, and A. Mendez-Vasquez. Sensory data set description language specification. Technical Report SDDL_Specification_v1.0, University of Florida, 2009.

[69] International Standards Organisation (ISO). *ISO-8601: Data elements and interchange formats – Information interchange – Representation of dates and times.* ISO, Geneva, Switzerland, 2004.

[70] K. Janowicz, A. Bröring, C. Stasch, S. Schade, T. Everding, and A. Llaves. A RESTful Proxy and Data Model for Linked Sensor Data. *International Journal of Digital Earth*, pages 1–20, Jan. 2011.

[71] C. Jiang and P. Steenkiste. A hybrid location model with a computable location identifier for ubiquitous computing. In *Proceedings of UbiComp '02*, pages 246–263, Gothenberg, Sweden, 2002. Springer-Verlag.

[72] M. Kabir, J. Han, J. Yu, and A. Colman. Scims: A social context information management system for socially-aware applications. In *Advanced Information Systems Engineering*, volume 7328 of *Lecture Notes in Computer Science*, pages 301–317. Springer Berlin Heidelberg, 2012.

[73] L. Kagal, T. Finin, and A. Joshi. A policy based approach to security for the semantic web. In *Proceedings of the second International Semantic Web Conference (ISWC 2003)*, pages 402–418, Sanibel Island, Florida, USA, 2003.

[74] R. Keskisrkk and E. Blomqvist. Semantic complex event processing for social media monitoring ? a survey. In *Proceedings of the international workshop on Social Media and Linked Data for Emergency Response (SMILE 2013)*, 2013.

[75] C. Koehler, B. D. Ziebart, J. Mankoff, and A. K. Dey. Therml: Occupancy prediction for thermostat control. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '13, pages 103–112, New York, NY, USA, 2013. ACM.

[76] N. Kourtellis, J. Finnis, P. Anderson, J. Blackburn, C. Borcea, and A. Iamnitchi. Prometheus: User-controlled p2p social data management for socially-aware applications. In I. Gupta and C. Mascolo, editors, *Middleware 2010*, volume 6452 of *Lecture Notes in Computer Science*, pages 212–231. Springer Berlin Heidelberg, 2010.

[77] R. A. Kowalski and M. J. Sergot. A logic-based calculus of events. *New Generation Comput.*, 4(1):67–95, 1986.

[78] D. Le-Phuoc, M. Dao-Tran, J. X. Parreira, and M. Hauswirth. A native and adaptive approach for unified processing of linked streams and linked data. In *Proceedings of the 10th international conference on The semantic web - Volume Part I*, ISWC'11, pages 370–388, Berlin, Heidelberg, 2011. Springer-Verlag.

[79] D. Le-Phuoc, M. Dao-Tran, J. X. Parreira, and M. Hauswirth. A native and adaptive approach for unified processing of linked streams and linked data. In *Proceedings of the 10th International Conference on The Semantic Web - Volume Part I*, ISWC'11, pages 370–388, Berlin, Heidelberg, 2011. Springer-Verlag.

[80] D. B. Lenat and R. V.Guha. *Building Large Knowledge-Based Systems; Representation and Inference in the Cyc Project.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.

[81] P. Lilis, I. Lourdi, C. Papatheodorou, and M. Gergatsoulis. A metadata model for representing time-dependent information in cultural collections. In *Proceedings of the first online metadata and semantics research conference (MTSR' 05)*, pages 1–12. Rinton Press, Nov. 2005.

[82] M. Luther, Y. Fukazawa, M. Wagner, and S. Kurakake. Situational reasoning for task-oriented mobile service recommendation. *The Knowledge Engineering Review*, 23(01):7–19, 2008.

[83] C. Lutz, F. Wolter, and M. Zakharyaschev. Temporal description logics: A survey. In *Proceedings of the 2008 15th International Symposium on Temporal Representation and Reasoning*, TIME '08, pages 3–14, Washington, DC, USA, 2008. IEEE Computer Society.

[84] B. McBride. Jena: A semantic web toolkit. *IEEE Internet Computing*, 6(6):55–59, 2002.

[85] B. McBride and M. Butler. Representing and querying historical information in RDF with application to E-discovery. Technical Report HPL-2009-261, Hewlit Packard Laboratories, 2009.

[86] M. Mendler and S. Scheele. Towards a type system for semantic streams. In *Proc. 1st Intl Workshop Stream Reasoning*, 2009.

[87] L. Moreau, B. Clifford, J. Freire, J. Futrelle, Y. Gil, P. Groth, N. Kwasnikowska, S. Miles, P. Missier, J. Myers, B. Plale, Y. Simmhan, E. Stephan, and J. V. den Bussche. The Open Provenance Model core specification (v1.1). *Future Generation Computer Systems*, July 2010.

[88] T. Moser, H. Roth, S. Rozsnyai, R. Mordinyi, and S. Biffl. Semantic event correlation using ontologies. In *Proceedings of the Confederated International Conferences, CoopIS, DOA, IS, and ODBASE 2009 on On the Move to Meaningful Internet Systems: Part II*, OTM '09, pages 1087–1094, Vilamoura, Portugal, 2009. Springer-Verlag.

[89] B. Motik, B. Cuenca Grau, and U. Sattler. Structured objects in owl: Representation and reasoning. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 555–564, New York, NY, USA, 2008. ACM.

[90] B. Motik and R. Rosati. Reconciling description logics and rules. *J. ACM*, 57(5):30:1–30:62, June 2008.

[91] B. Motik, U. Sattler, and R. Studer. Query Answering for OWL-DL with rules. *J. Web Sem.*, 3(1):41–60, 2005.

[92] B. Motik, R. Shearer, and I. Horrocks. Hypertableau reasoning for description logics. *J. Artif. Int. Res.*, 36(1):165–228, Sept. 2009.

[93] R. Nevatia, J. Hobbs, and B. Bolles. An ontology for video event representation. In *Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04) Volume 7*, CVPRW '04, pages 119–129, Washington, DC, USA, 2004. IEEE Computer Society.

[94] T. P. Nogueira, R. B. Braga, and H. Martin. An ontology-based approach to represent trajectory characteristics. In *Computing for Geospatial Research and Application (COM.Geo), 2014 Fifth International Conference on*, pages 102–107, Aug 2014.

[95] N. F. Noy, R. W. Fergerson, and M. A. Musen. The knowledge model of Protége-2000: Combining interoperability and flexibility. In *Proceedings of the 2th International Conference on Knowledge Engineering and Knowledge Management (EKAW' 2000)*, pages 17–32, Juan-les-Pins, France, 2000. Springer.

[96] M. J. O'Connor and A. K. Das. SQWRL: A query language for OWL. In *OWL: Experiences and Directions (OWLED)*, volume 529 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2009.

[97] M. J. O'Connor and A. K. Das. A lightweight model for representing and reasoning with temporal information in biomedical ontologies. In *Proceedings of the International Conference on Health Informatics(HEALTHINF 2010)*, Valencia, Spain, 2010.

[98] F. Orlandi and A. Passant. Modelling provenance of dbpedia resources using wikipedia contributions. *J. Web Sem.*, 9(2):149–164, 2011.

[99] J. Ouaknine and J. Worrell. Some recent results in metric temporal logic. *Formal Modeling and Analysis of Timed Systems*, pages 1–13, 2008.

[100] F. Paganelli and D. Giuli. An ontology-based system for context-aware and configurable services to support home-based continuous care. *Information Technology in Biomedicine, IEEE Transactions on*, 15(2):324 –333, March 2011.

[101] A. Paschke, H. Boley, and P. Vincent. Semantic complex event processing – the future of dynamic IT. Invited talk at SemTech 2010, June 2010.

[102] T. Patkos, I. Chrysakis, A. Bikakis, D. Plexousakis, and G. Antoniou. A reasoning framework for ambient intelligence. In *Proceedings of the 6th Hellenic conference on Artificial Intelligence: theories, models and applications*, SETN'10, pages 213–222, Athens, Greece, 2010. Springer-Verlag.

[103] I. S. E. Peraldi, A. Kaya, and R. Möller. Formalizing multimedia interpretation based on abduction over description logic ABoxes. In *Description Logics*, volume 477 of *CEUR Workshop Proceedings*, 2009.

[104] F. Perich, A. Joshi, T. Finin, and Y. Yesha. On data management in pervasive computing environments. *IEEE Trans. on Knowl. and Data Eng.*, 16(5):621–634, May 2004.

[105] M. Perry, A. Sheth, and P. Jain. SPARQLST:Extending SPARQL to Support Spatiotemporal Queries. Technical Report KNOESIS-TR-2009-01, Kno.e.sis Center Technical Report, 2009.

[106] A. Pugliese, O. Udrea, and V. S. Subrahmanian. Scaling RDF with Time. In *Proceeding of the 17th international conference on World Wide Web (WWW '08)*, pages 605–614, Beijing, China, 2008. ACM.

[107] A. Ranganathan, J. Al-Muhtadi, and R. H. Campbel. Reasoning about uncertain contexts in pervasive computing environments. *IEEE Pervasive Computing*, 3(2):62–70, 2004.

[108] A. Ranganathan, J. Al-Muhtadi, S. Chetan, R. Campbell, and M. D. Mickunas. Middlewhere: a middleware for location awareness in ubiquitous computing applications. In *Proceedings of Middleware '04*, pages 397–416, New York, USA, 2004. Springer-Verlag New York, Inc.

[109] K. Rehrl, S. Leitinger, S. Krampe, and R. Stumptner. An approach to semantic processing of gps traces. In *MPA'10: Proceedings of the 1st Workshop on Movement Pattern Analysis*, September 2010.

[110] D. Riboni and C. Bettini. Context-aware activity recognition through a combination of ontological and statistical reasoning. In *Proceedings of the 6th International Conference on Ubiquitous Intelligence and Computing*, UIC '09, pages 39–53, Berlin, Heidelberg, 2009. Springer-Verlag.

[111] D. Riboni and C. Bettini. Cosar: hybrid reasoning for context-aware activity recognition. *Personal and Ubiquitous Computing*, 15(3):271–289, 2011.

[112] D. Riboni and C. Bettini. OWL 2 modeling and reasoning with complex human activities. *Pervasive and Mobile Computing*, 7(3):379–395, 2011.

[113] D. Riboni and C. Bettini. Context provenance to enhance the dependability of ambient intelligence systems. *Personal and Ubiquitous Computing*, 16(7):799–818, 2012.

[114] N. D. Rodriguez, M. P. Cuellar, J. Lilius, and M. D. Calvo-Flores. A fuzzy ontology for semantic modelling and recognition of human behaviour. *Knowledge-Based Systems*, 66(0):46 – 60, 2014.

[115] M. Roman, C. K. Hess, R. Cerqueira, A. Ranganathan, R. H. Campbell, and K. Nahrstedt. Gaia: A Middleware Infrastructure to Enable Active Spaces. *IEEE Pervasive Computing*, pages 74–83, Oct–Dec 2002.

[116] R. Rosati. DL+log: Tight integration of description logics and disjunctive datalog. In P. Doherty, J. Mylopoulos, and C. A. Welty, editors, *Proceedings of the tenth International Conference on Principles of Knowledge Representation and Reasoning*, pages 68–78. AAAI Press, 2006.

[117] A. Rosi, S. Dobson, M. Mamei, G. Stevenson, J. Ye, and F. Zambonelli. Social sensors and pervasive services: Approaches and perspectives. In *PerCol2011: Proceedings of the Second International Workshop on Pervasive Collaboration and Social Networking*, March 2011.

[118] S. Rozsnyai, R. Vecera, J. Schiefer, and A. Schatten. Event cloud - searching for correlated business events. In *IEEE International Conference on E-Commerce Technology and Enterprise Computing, E-Commerce, and E-Services*, volume 0, pages 409–420, Los Alamitos, CA, USA, 2007. IEEE Computer Society.

[119] D. Russomanno, C. Kothari, and O. Thomas. Sensor ontologies: From shallow to deep models. In *Proceedings of the*

37th Southeastern Symposium on System Theory (SSST '05), pages 107–112, Mar. 2005.

[120] K. Sagonas, T. Swift, and D. S. Warren. Xsb as an efficient deductive database engine. In *In Proceedings of the ACM SIGMOD International Conferenceon the Management of Data*, pages 442–453. ACM Press, 1994.

[121] S. Santini, B. Ostermaier, and A. Vitaletti. First experiences using wireless sensor networks for noise pollution monitoring. In *Proceedings of the workshop on Real-world wireless sensor networks*, REALWSN '08, pages 61–65, Glasgow, Scotland, 2008. ACM.

[122] M. Shanahan. Perception as abduction: Turning sensor data into meaningful representation. *Cognitive Science*, 29(1):103–134, 2005.

[123] A. Sheth, C. Henson, and S. S. Sahoo. Semantic sensor web. *Internet Computing*, 12(4), July/August 2008.

[124] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz. Pellet: A practical OWL-DL reasoner. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(2):51–53, 2011.

[125] S. Spaccapietra, C. Parent, M. L. Damiani, J. A. de Macedo, F. Porto, and C. Vangenot. A conceptual view on trajectories. *Data Knowl. Eng.*, 65(1):126–146, Apr. 2008.

[126] T. Springer and A.-Y. Turhan. Employing description logics in ambient intelligence for modeling and reasoning about complex situations. *JAISE*, 1(3):235–259, 2009.

[127] M. Stabeler, G. Stevenson, S. Dobson, and P. Nixon. Basadaeir: harvesting user profiles to bootstrap pervasive applications. In *Proceedings of the 7th International Conference on Pervasive Computing, Pervasive 2009.*, Nara, Japan, May 2009.

[128] G. Stevenson and S. Dobson. Sapphire: Generating Java runtime artefacts from OWL ontologies. In *Proceedings of the 3rd International Workshop on Ontology-Driven Information Systems Engineering (ODISE 2011)*, pages 425–436, London, UK, 2011.

[129] G. Stevenson, S. Knox, S. Dobson, and P. Nixon. Ontonym: a collection of upper ontologies for developing pervasive systems. In *Proceedings of the 1st Workshop on Context, Information and Ontologies*, CIAO '09, pages 9:1–9:8, Heraklion, Greece, 2009. ACM.

[130] G. Stevenson, J. Ye, and S. Dobson. On the impact of the temporal features of sensed data on the development of pervasive systems. In *Proceedings of the International Workshop on Programming Methods for Mobile and Pervasive Systems (PMMPS' 10)*, Helsinki, Finland, May 2010.

[131] G. Stevenson, J. Ye, S. Dobson, and P. Nixon. Loc8: A location model and extensible framework for programming with location. *IEEE Pervasive Computing*, 9(1):28 – 37, January 2010.

[132] G. Stoilos, G. B. Stamou, J. Z. Pan, V. Tzouvaras, and I. Horrocks. Reasoning with very expressive fuzzy description logics. *J. Artif. Intell. Res. (JAIR)*, 30:273–320, 2007.

[133] U. Straccia. Managing uncertainty and vagueness in description logics, logic programs and description logic programs. In C. Baroglio, P. A. Bonatti, J. Maluszyński, M. Marchiori, A. Polleres, and S. Schaffert, editors, *Reasoning Web*, pages 54–103. Springer-Verlag, Berlin, Heidelberg, 2008.

[134] T. Strang, C. Linnhoff-Popien, and K. Frank. Applications of a Context Ontology Language. In D. Begusic and N. Rozic, editors, *Proceedings of the International Conference on Software, Telecommunications and Computer Networks (SoftCom2003)*, pages 14–18. Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, University of Split, Croatia, October 2003.

[135] H. Stuckenschmidt, S. Ceri, E. Della Valle, F. Van Harmelen, and P. di Milano. Towards expressive stream reasoning. In *Proceedings of the Dagstuhl Seminar on Semantic Aspects of Sensor Networks*, 2010.

[136] Sensor Web Enablement group. `http://www.opengeospatial.org/projects/groups/sensorwebdwg`.

[137] J. Tappolet and A. Bernstein. Applied Temporal RDF: Efficient Temporal Querying of RDF Data with SPARQL. In *Proceedings of the 6th European Semantic Web Conference on The Semantic Web*, pages 308–322, Heraklion, Crete, Greece, 2009. Springer-Verlag.

[138] K. Taylor and L. Leidinger. Ontology-driven complex event processing in heterogeneous sensor networks. In *Proceedings of the 8th extended semantic web conference on The semantic web: research and applications - Volume Part II*, pages 285–299, Heraklion, Crete, Greece, 2011. Springer-Verlag.

[139] H. J. ter Horst. Extending the rdfs entailment lemma. In *International Semantic Web Conference*, pages 77–91, 2004.

[140] K. Teymourian and A. Paschke. Semantic rule-based complex event processing. In *Proceedings of the 2009 International Symposium on Rule Interchange and Applications*, RuleML '09, pages 82–92, Las Vegas, Nevada, 2009. Springer-Verlag.

[141] Y. Theoharis, I. Fundulaki, G. Karvounarakis, and V. Christophides. On provenance of queries on semantic web data. *IEEE Internet Computing*, 15(1):31–39, Jan. 2011.

[142] A. Toninelli, R. Montanari, L. Kagal, and O. Lassila. A semantic context-aware access control framework for secure collaborations in pervasive computing environments. In *Proceedings of the 5th international conference on The Semantic Web*, ISWC'06, pages 473–486, Athens, GA, 2006. Springer-Verlag.

[143] A. Toninelli, A. Pathak, and V. Issarny. Yarta: A middleware for managing mobile social ecosystems. In *Proceedings of the 6th International Conference on Advances in Grid and Pervasive Computing*, GPC'11, pages 209–220, Berlin, Heidelberg, 2011. Springer-Verlag.

[144] D. Tsarkov and I. Horrocks. FaCT++ Description Logic Reasoner: System Description. In *International Joint Conference on Automated Reasoning (IJCAR 2006)*, pages 292–297, 2006.

[145] M. Vacura, V. Svátek, and P. Smrž. A pattern-based framework for uncertainty representation in ontologies. In *Proceedings of the 11th international conference on Text, Speech and Dialogue*, TSD '08, pages 227–234, Brno, Czech Republic, 2008. Springer-Verlag.

[146] W. R. van Hage, V. Malaise, R. H. Segers, L. Hollink, and G. Schreiber. Design and use of the simple event model (sem). *Web Semantics: Science, Services and Agents on the World Wide Web*, 9(2), 2011.

[147] M. Vökel. RDFReactor – from ontologies to programmatic data access. In *The Jena Developer Conference*, Bristol, UK, 2006.
[148] M. Völkel and Y. Sure. RDFReactor - From Ontologies to Programmatic Data Access. In *Poster and Demo proceedings fo the International Semantic Web Conference (ISWC) 2005, Galway, Ireland*, Nov. 2005.
[149] X. Wang, J. S. Dong, C.-Y. Chin, S. Hettiarachchi, and D. Zhang. Semantic Space: An Infrastructure for Smart Spaces. *IEEE Pervasive Computing*, 3(3):32–39, 2004.
[150] W. Wei and P. Barnaghi. Semantic annotation and reasoning for sensor data. In *Proceedings of the 4th European conference on Smart sensing and context*, EuroSSC'09, pages 66–76, Berlin, Heidelberg, 2009. Springer-Verlag.
[151] T. W. Wlodarczyk, C. Rong, M. O'Connor, and M. Musen. SWRL-F: a fuzzy logic extension of the semantic web rule language. In *Proceedings of the International Conference on Web Intelligence, Mining and Semantics*, WIMS '11, pages 39:1–39:9, Sogndal, Norway, 2011. ACM.
[152] Z. Yan, D. Chakraborty, C. Parent, S. Spaccapietra, and K. Aberer. Semantic trajectories: Mobility data computation and annotation. *ACM Trans. Intell. Syst. Technol.*, 4(3):49:1–49:38, July 2013.
[153] J. Ye, L. Coyle, S. Dobson, and P. Nixon. Ontology-based models in pervasive computing systems. *The Knowledge Engineering Review*, 22(04):315–347, December 2007.
[154] J. Ye, S. Dobson, and S. McKeever. Situation identification techniques in pervasive computing: a review. *Pervasive and mobile computing*, 8:36–66, Feb. 2012.
[155] J. Ye, G. Stevenson, and S. Dobson. Usmart: an unsupervised semantic mining activity recognition technique. *ACM Transactions on Interactive Intelligent Systems*, 2014. To appear.
[156] J. Ye, G. Stevenson, S. Dobson, M. O'Grady, and G. O'Hare. $\mathcal{PI}$: Perceiver and interpreter of smart home datasets. In *Proceedings of the 5th International ICST Conference on Pervasive Computing Technologies for Healthcare (Pervasive-Health 2011)*, pages 131–138, Dublin, Ireland, 2011.
[157] Z. Yu, X. Zhou, Z. Yu, J. H. Park, and J. Ma. iMuseum: A scalable context-aware intelligent museum system. *Computer Communications*, 31(18):4376–4382, 2008.
[158] W. Zhang, K. M. Hansen, and T. Kunz. Enhancing intelligence and dependability of a product line enabled pervasive middleware. *Pervasive and Mobile Computing*, 6(2):198–217, 2010.
[159] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma. Mining interesting locations and travel sequences from GPS trajectories. In *Proceedings of the international World Wide Web conference (WWW 2009)*, pages 791–801, Madrid, Spain, 2009.